Third Anniversary Issue

Antic®

The ATARI® Resource

MAY 1985 VOLUME 4, NUMBER 1

U.S.A. $3.50
CANADA $4.00

The New Super Ataris



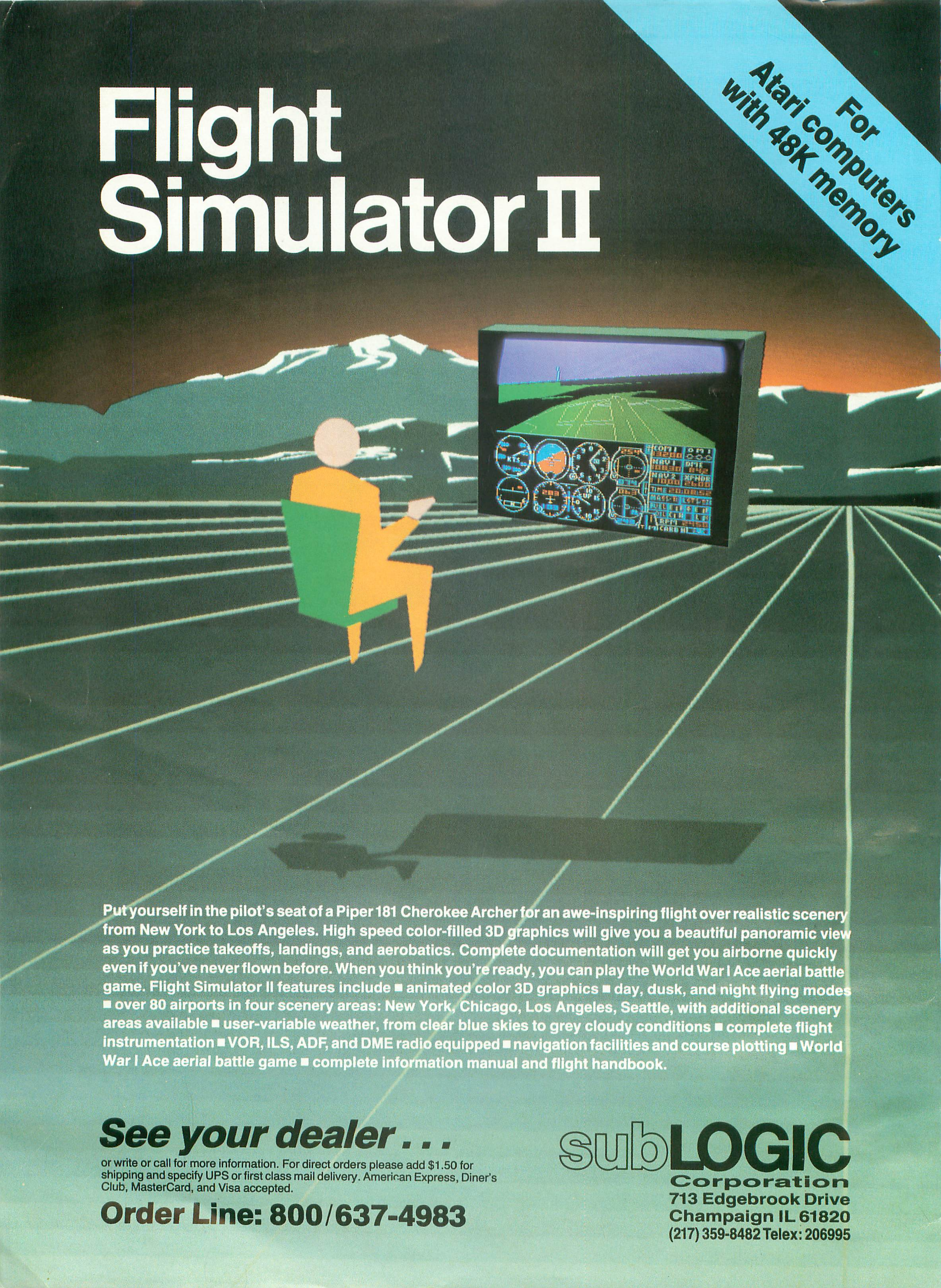
68000—Mighty 512K Atari "Brain"

GEM—New "Color Mac" Atari Power

State-of-the-art word processor!

Inside: 10 Type-in programs

0 74470 12728 05


Flight Simulator II
For Atari computers with 48K memory
Put yourself in the pilot's seat of a Piper 181 Cherokee Archer for an awe-inspiring flight over realistic scenery from New York to Los Angeles. High speed color-filled 3D graphics will give you a beautiful panoramic view as you practice takeoffs, landings, and aerobatics. Complete documentation will get you airborne quickly even if you've never flown before. When you think you're ready, you can play the World War I Ace aerial battle game. Flight Simulator II features include ■ animated color 3D graphics ■ day, dusk, and night flying modes ■ over 80 airports in four scenery areas: New York, Chicago, Los Angeles, Seattle, with additional scenery areas available ■ user-variable weather, from clear blue skies to grey cloudy conditions ■ complete flight instrumentation ■ VOR, ILS, ADF, and DME radio equipped ■ navigation facilities and course plotting ■ World War I Ace aerial battle game ■ complete information manual and flight handbook.
See your dealer . . .
or write or call for more information. For direct orders please add $1.50 for shipping and specify UPS or first class mail delivery. American Express, Diner's Club, MasterCard, and Visa accepted.
Order Line: 800/637-4983
subLOGIC
Corporation
713 Edgebrook Drive
Champaign IL 61820
(217) 359-8482 Telex: 206995

SURVIVAL OF THE SMARTEST . . .

Outsmart your Friends. Outwit the Dragon. Join the Quest.

Here are just a few of many screens.



48K Atari Disk

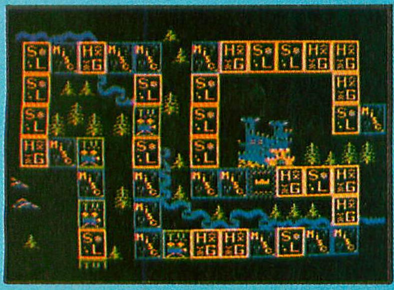
$39.95

Challenging excitement for 1 to 4 adventurers

THE CHALLENGE.
Over 2000 stimulating trivia/fact questions will send you rummaging through your personal memory banks. Answer correctly and advance in your quest while adding more gold to your treasure. Fail and you lose ground.



THE ACTION.
You may have to face and battle a ferocious dragon. If you prevail, the dragon's cache of gold will be added to your treasure. If you fail, your journey will be slowed while you heal your wounds.



THE STRATEGY
Each questing party consists of three characters, a Knight, Prince and Page. Each has distinctly different physical characteristics, and all must complete the Quest. You must decide when to send them on to the next challenge.



This exciting new concept in computer entertainment will keep you and your friends involved for hours and hours. This is probably the best party-game ever developed, and new question disks will be available.
Optional: Utility disk which allows you to create unlimited trivia questions and answers for educational or entertainment. The utility disk also includes over 1000 additional questions. **Utility disk $24.95.**

Royal Software

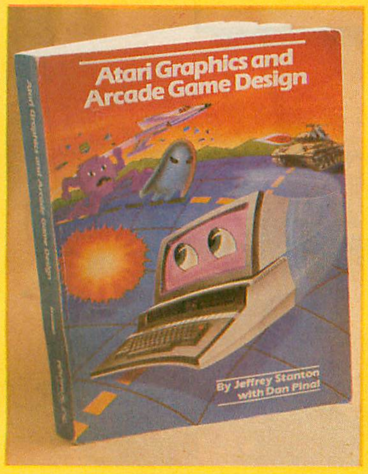
"Software fit for a king!"
2160 W. 11th • Eugene, OR 97402 • (503) 683-5361

Ask for Trivia Quest at your favorite Atari Computer Store or order directly from Royal Software. Use your MasterCard, Visa, American Express, or send check or Money Order including $2.90 shipping and handling.

Antic®

The ATARI® Resource

MAY 1985, VOLUME 4, NUMBER 1



Latest reviews of Logo and graphics . . 16



Thumbs up for Hitchhikers 19



Atari judges the battle of the brews . . 43

FEATURES

DEPARTMENTS

SOFTWARE LIBRARY

Antic
The ATARI® Resource

Publisher
James Capparell

Editorial Department
Nat Friedland, Editor; Jack Powell, Technical Editor; Michael Ciraolo, Associate Editor; Charles Jackson, Program Editor; Ron Luks, On-Line Editor.

Contributing Editors
Carl Evans, Ken Harms, Jerry White, Suzi Subeck, Anita Malnig.

Art Department
Marni Tapscott, Art Director; Diane Lindley, Production Supervisor; Linda Tapscott, Ad Production Coordinator; Patricia Fostar, Production Assistant.

Cover Photograph
Erik Weber

Cover Illustration
Kathy Riethmeier

Circulation Department
Les Torok, Manager; Hun-sik Kim, Shipping; Monica Burrell, Subscriptions; Eve Gowdey, Dealer Sales; Doug Millison, Dealer Sales; Brandt/Klingle, Circulation Consultants.

Accounting Department
V.J. Briggs, Manager; Brenda Oliver, Accounts Receivable; Lorene Kaatz, Credit Manager; Andrew Pope, Customer Service, Retailers.

Marketing
Gary Yost, Manager, Marketing Services; Steve Randall, Advertising Sales Director; Harvey Bernstein, Advertising Sales; Garland Associates, East Coast Representative.

Maria E. Chavez, Receptionist

General Offices (415) 957-0886
Advertising Sales (415) 661-3400
Credit Card Subscriptions
outside California (800) 227-1617 ext. 133
inside California (800) 772-3545 ext. 133
Subscription Problems (415) 397-1881

May 1985, Volume 4, Number 1.
Antic—The Atari Resource is published twelve times per year by Antic Publishing. Editorial offices are located at 524 Second Street, San Francisco, CA 94107. ISSN 0745-2527. Second Class Postage paid at San Francisco, California and additional mailing offices. POSTMASTER: Send address change to **Antic**, 524 Second Street, San Francisco, CA 94107.

Editorial submissions should include program listing on disk or cassette, and text file on media and paper if text was prepared with a word processor. Media will be returned if self-addressed stamped mailer is supplied. **Antic** assumes no responsibility for unsolicited editorial material.

editorial

Erik Weber



Top row, left to right: Andrew Pope, Hun-sik Kim, Lorene Kaatz, Diane Lindley, Doug Millison; Second row from top, left to right: Maria Chavez, Gary Yost, Michael Ciraolo, Brenda Oliver, Marni Tapscott, James Capparell; Second row from bottom, left to right: V.J. Briggs, Jack Powell, Linda Tapscott, Charles Jackson; Bottom row, left to right: Les Torok, Steve Randall, Nat Friedland, Harvey Bernstein. Not pictured: Eve Gowdey, Patricia Fostar, Monica Burrell.

With this issue of **Antic**, the magazine starts its fourth year of publication.

The early issues were put out from the kitchen table of a former NASA programmer who had founded ABACUS, the San Francisco Atari Users Group. And the magazine rocketed to

continued on page 8


GHOSTBUSTERS™
THE COMPUTER GAME
BY DAVID CRANE
NOW AVAILABLE FOR
COMMODORE 64™, APPLE II AND
ATARI® HOME COMPUTERS
NOW PLAYING ON A COMPUTER NEAR YOU...
DON'T MISS IT!
CITY'S PK ENERGY: 1162 $1000
CITY'S PK ENERGY: 1478 $2500
CITY'S PK ENERGY: 3999 $48800
Available on disk.
ACTIVISION
HOME COMPUTER SOFTWARE™
For information about new releases, call:
(800) 633-4263 anytime on weekends.
In California, call (415) 940-6044.
COMMODORE 64™ IS A TRADEMARK OF COMMODORE ELECTRONICS LTD. GHOSTBUSTERS™ IS A TRADEMARK OF COLUMBIA PICTURES INDUSTRIES, INC. GHOSTBUSTERS LOGO © 1985 COLUMBIA PICTURES INDUSTRIES, INC. ALL RIGHTS RESERVED. APPLE II IS A REGISTERED TRADEMARK OF APPLE COMPUTER. ATARI® IS A TRADEMARK OF ATARI, INC. © 1985 ACTIVISION, INC.

EDITORIAL
continued from page 6

a 120-page monthly with over 100,000 circulation—almost before we had time to stop keeping our records on scraps of paper stuffed into shoeboxes.

In many ways, 1984 was **Antic's** most challenging year. The microcomputer shakeout hit the market hard. Suddenly many computer stores wouldn't accept Atari products and a number of our steadiest advertisers went out of business. Some of the best-established computer magazines went under during 1984. Frankly, there were computer business "experts" who wouldn't have been surprised to see **Antic** call it quits too. . .

But the **Antic** staff is just too innovative and dedicated to ever give in to a downturn. We simply tightened our belts and looked for more ways to work smarter. And now we've come through stronger than ever—as the turnaround of the past few months vindicated all those who believed in the Atari as the best 8-bit personal computer ever made.

New subscriptions have been pouring into **Antic** at the rate of nearly 1,000 a week. More and more third-party manufacturers who'd turned their backs on Atari are now coming back to show their support in the pages of this magazine. Even more important, **Antic** has lived up to the pledge we made to our readers last autumn—to find new ways to fill the vacuum in Atari information and services. . .

You can now read full details of the latest Atari news just hours after it happens, in the ANTIC ONLINE edition on CompuServe. And in only a few short months, the Antic Arcade catalog has become one of the most important outlets for top-quality Atari books and software—including many of the previously out-of-print APX software classics.

These are only two of the new services **Antic** began providing this year; there's also the national directory of authorized Atari service centers, the Worldwide Users Group Network (WUN). . .and more!

So the **Antic** Third Anniversary arrives right in the middle of our most productive and exciting period ever. Thanks for coming along with us. **Antic** will have even bigger and better surprises for you during the rest of 1985!

And you can bet that the best coverage of the new Atari XE and ST computer models will continue to be found right here in these pages!

. . .and we won't take it anymore!

Dear **Antic**
Attached is a copy of the letter I sent to Broderbund Software expressing my disappointment in their decision not to publish an Atari edition of **Championship Loderunner**.

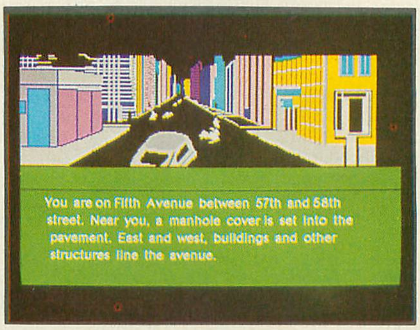
I think all Atari owners should unite and start writing letters to software companies to let them know how many of us are out there.

Timothy F. Hitchings
Staten Island, NY

Mr. Timothy F. Hitchings is absolutely right. . .and **Antic** has received many letters similar to the one above. It's time for Atari owners to *demand* first-class citizenship in the personal computer software world!

Now is the time to take action, because there is no longer the slightest excuse for major software companies to avoid bringing out their hit products for the Atari.

Just about half a million Atari 800XLs were sold during the 1984 holiday season. In many stores, Atari 1050 disk drives moved out just as fast—as previous owners upgraded to disk.



Fahrenheit 451

Add this to nearly a million previously-sold compatible Atari computers and you have a vast user base that does not have to take second place anywhere in the personal computer market.

So when YOU see a piece of software you'd like to buy, and you learn it isn't being released for the Atari, your next step should be to write a letter to the president of the software company explaining what a big mistake they are making.

You can usually find the company's address on the software package or on the advertisement for the product. You don't need to look up the name of the person who heads the company—just write PRESIDENT on the envelope above the company name and address. Feel free to enclose a photocopy of this editorial along with your letter, in order to add a second voice to your argument.

On my desk at **Antic** as I write this, there's a pile of superbly packaged color-graphics adventure software from Spinnaker—all for the Apple and Commodore computers.

Spinnaker's Teralium line (originally called Trillium) features graphic adventure software adapted from famous science fiction books including Ray Bradbury's *Farenheit 451*,

continued on page 10

LOTSABYTES CONTINUES THE WAR!

WAR on high prices! We're going to put an end to the software price 'ripoff'. And **YOU** can help! Just keep those orders coming while you continue to enjoy the **quality, quantity, selection** and **low prices** that you deserve. Our National Public Domain Copy Service will save you time, tedious work, and money. And our **exclusive** distribution of **sharply discounted** commercial programs will bring you some of the finest programs for the lowest possible price, usually 50% and more off retail! You continue to get **FREE BONUSES** with each purchase of three or more disks.

PUBLIC DOMAIN SOFTWARE

#1 **GAMES** Two full disk sides packed with over 25 games including some Arcade quality. **$7.95**	#2 **UTILITIES** 25 powerful programs to help you get the most out of your Atari computer. **$7.95**	#3 **AMS MUSIC** 25 Advanced Musicsystem files including a new Player program. 2 sides. **$7.95**	#4 **GAMES** All different! 14 more better games on 2 disk sides. Some Arcade types. **$7.95**	#5 **EDUCATION** Loaded with 28 programs on 2 disk sides Fun learning for the whole family. **$7.95**
#6 **AMS MUSIC** 25 all-time favorites with a Player program. Two sides. **$7.95**	#7 **GAMES** Two disk sides packed with 14 more great games. Some Arcade types. **$7.95**	#8 **UTILITIES** 17 more power-packed utilities to help unleash full potential of your Atari. **$7.95**	#9 **GAMES** Two full sides filled with 17 of the best and most recent. Some Arcade. **$7.95**	#10 **UTILITIES** A new assortment of 17 great and powerful programs Don't miss it! **$7.95**
#11 **GAMES** *NEW!* Our newest. 2 sides filled with great games. **$7.95**	#12 **ADVENTURES** *NEW!* 2 full disk sides filled with text adventures. **$7.95**	#13 **EDUCATION** *NEW!* 2 disk sides filled with something for everyone. **$7.95**	#14 **AMS MUSIC** *NEW!* 2 sides filled with great music and a player program. **$7.95**	#15 **UTILITIES** *NEW!* Another assortment of fine programs. Not to be missed. **$7.95**

LotsaBytes *EXCLUSIVES*

ADVANCED MUSICSYSTEM II
by LEE ACTOR

Allows you to create music with your Atari computer! All new machine code.



* Control over pitch duration, envelope dynamic level, meter, tempo and key.
* 4 independent voices
* 5½ octaves per voice
* Save up to 8200 notes
* Custom DOS
* FULL instructions
* 24K disk

Originally $29.95 **Only $14.95**

ORIGINAL ADVENTURE
by Bob Howell

For all Atari computers. The Original Colossal Cave Adventure faithfully reproduced from the 'main-frames'. This is the one that launched the whole Adventure craze of today!



* Two mazes
* 130 rooms
* Deadly Dragons
* Nasty Dwarves
* Tenacious Troll
* The Pirate & More!
* 86 coded hints
* SAVE RESUME
* 40K disk or 32K tape

Originally $24.95 **Only $14.95**

QUALITY WORD PROCESSING

ESI WRITER! At last a brand-new Word Processor that has more features and is easier to use than anything else available for the Atari. Easy for the beginner to use, it asks questions and remembers the answers. ESI WRITER is so sophisticated that it has many more features we don't even have room to mention! Works with ANY Atari.

* Reads any text file * Built in Help screen
* Very fast! * Works with ANY printer
* Instant top, bottom or text location without scrolling!
* Every printer feature * DISK ONLY (Any Atari)
* Search and replace * Block move text
* Page eject/start * Set margins/lines etc.
* Full justification * Print headers etc.
* Block delete etc. * Change video color
* Over 50 pages of docs and tutorials

TRUST US ON THIS ONE! YOU WILL LOVE IT!

Originally $49.95 **LotsaBytes price $19.95**

** * FREE BONUSES * **

Now for each 3 disks ordered you may choose any 1 of the following disks FREE!!

... buy 3 - get 1, buy 6 - get 2, buy 9 get 3 ...

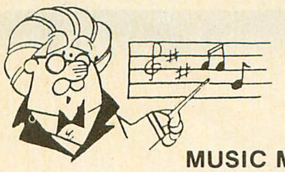
a. The Atari XL TRANSLATOR DISK that enables XL owners to use most 400/800 software. **FREE!!**

-- or --

b. An all different AMS MUSIC disk with Player. **FREE!!**

--or --

c. Your choice of one of the P.D. disks -- #1, #2, #3, #4, #5, #6, #7, #8, #9, or #10 (specify one) **FREE!!**



MUSIC MAJOR!

Learn the basics of music with this light-hearted but very thorough approach. Covering such topics as note recognition, key signatures, note counting, and much more, it is designed for use by both the individual student and music class.

This program includes a thoroughly illustrated manual and offers a QUIZ MASTER utility that allows the teacher or the self-taught student to create their own A-B-C-D type tests, with a sample quiz included.

Originally $39.95 ***Only $14.95***

* * * *

GREAT GAMES!

SPACE GAMES: Three games for **one** low price!. In **Aliens** you can't get them all and the pace keeps getting faster. When you do get rid of most of them, you are left in a space quadrant peppered with mines. Will you **Survive**? If you do, you must penetrate the alien's spaceship, survive a **Robot Attack**, and get back your stolen 'cloaking' device! Interested?

$24.95 list ***LotsaBytes price:* $9.95**

THE BEAN MACHINE by Steve Robinson is an Award Winning Arcade game that will drive you crazy balancing a series of beams while trying to get all the beans to roll down, without touching, all the while avoiding 'strange creatures' who drop in to steal the beans. It's addicting!

$24.95 list ***LotsaBytes price:* $9.95**

DIGGERBONK, another Award Winning game by Steve Robinson, challenges you to find your way through a continuously scrolling maze while avoiding some really strange creatures. Along the way you will need to Bonk some of them, but watch out for the bombs.

$24.95 list ***LotsaBytes price:* $9.95**

GUESS WHAT'S COMING TO DINNER lets you try to manuever a snake through 7 levels if you can keep it from starving or being electrocuted. Lots of surprises! One or two players.

$24.95 list ***LotsaBytes price:* $9.95**

* * * *

CREATIVE LEARNING ADVENTURES
Ages 4 to 10 — Disk only

1. Hours of **educational** fun playing 3 exciting creative adventures with a friendly alien learning about our planet Earth. Hand/eye co-ordination, drawing, and music skills are emphasized.

$24.95 list ***LotsaBytes price:* $12.95**

2. Four challenging learning games that are the favorites of our friendly alien. Helps your child to develop logical reasoning ability.

$24.94 list ***LotsaBytes price:* $12.95**

3. These 3 Fun-Day learning games will help with intellectual development, hand/eye co-ordination, logic, spatial, and analytical abilities.

$24.95 list ***LotsaBytes price:* $12.95**

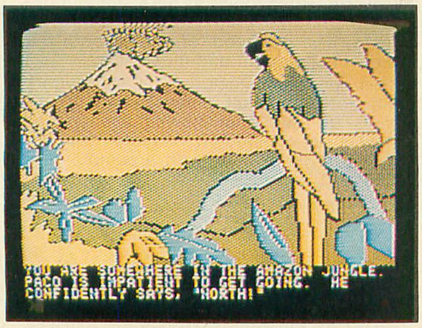
Full 100% Replacement guarantee.Any disk found to be defective will be replaced free and we will also refund your return postage. All orders shipped by First Class U.S. Mail. Add $1.95 shipping and handling for 1 to 5 disks. Add $2.95 for 6 to 12 disks. California residents add 6% sales tax. Outside of U.S.A. and Canada add 15%. U.S. Funds only. We accept checks or Money Orders. Sorry, no COD or Charge Cards. Allow three weeks for personal checks to clear.

LOTSABYTES

15445 Ventura Blvd., Suite 10G, Sherman Oaks, CA 91413

i/o board

EDITORIAL
continued from page 8



Amazon

Arthur C. Clarke's *Rendezvous With Rama* and Michael Crichton's *Amazon*.

The Windham Classics line from Spinnaker presents interactive graphic software adaptations of some of the best-known *children's* books. In this series are *Swiss Family Robinson*, *Treasure Island*, *Alice In Wonderland* and *Below The Root*.

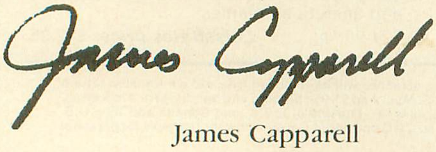
But at this writing, Spinnaker says it won't release any of these products for the Atari.

If this makes YOU a little angry, then it would be a good idea to write a letter to:

William H. Bowman
Spinnaker Software
One Kendall Square
Cambridge, MA 02139

Antic will continue to report on this situation until all important software companies stop short-changing Atari owners.

Please feel free to send **Antic** copies of your letters to software companies—and let us know about any responses you get from the companies. This will help us keep everybody informed about the latest victories and opposition in this ongoing struggle.



James Capparell
Publisher

MODIFICATIONS AND KUDOS FOR TYPO II

Congratulations on your error checking program, TYPO II. It is short, fast and a great help in accurately typing your listings.

Here are a couple of modifications. For those of us with BASIC XL or an auto-numbering program, typing in the line numbers is not considered a life enriching experience. The following changes let you step through each program line by typing an asterisk and [RETURN].

This causes each succeeding program line to be listed and automatically checked, making it possible to type the program with an auto-numbering routine, and then check it without ever typing in a line number.

Even if you don't use auto-numbering, these changes will make it a lot pleasanter for you to step through the lines of a previously typed program that you're modifying.

Line 32210 is changed to make the two-letter codes appear in white, as they do in the magazine.

```
HG 32025 X=PEEK(136)+PE
   EK(137)*256
FW 32065 IF LINE$="*" T
   HEN GOSUB 32230:POSI
   TION 2,4:LIST B:POKE
   764,12:GOTO 32060
FY 32210 POSITION 0,16:
   ? CHR$(HCODE+128);CH
   R$(LCODE+128)
EI 32230 B=PEEK(X)+PEEK
   (X+1)*256:IF B=32000
   THEN POP :GOTO B
FA 32240 X=X+PEEK(X+2):
   RETURN
```

Patrick Dell'Era
Fairfax, CA

HELP FOR PROGRAMS

How can I incorporate the "help" key on my Atari into my utility programs?

Greg Lyles
Rosemead, CA

To clear the HELP key, POKE 732,0. To read that key, PEEK(732). A 17 represents the HELP key, an 81 represents SHIFT-HELP, and a 145 means CONTROL-HELP.
—ANTIC ED

TRAK REPAIRS

When my Trak disk drive went on the blink, I was somewhat upset to find that the company had gone out of business. But after a call to Computer Palace in Oregon—where I bought my drive through an **Antic** mail-order ad—I found I could get my Trak repaired by Electronic Connexion, 424 E. Stroop Road, Kettering, Ohio 45429. You can phone them between 11–3 eastern time at (513) 294-0212.

Please print this information in your great magazine as a service to other Trak owners.

William R. Goslin
Grand Isle, LA

BEATING THOSE FOOTBALL BLUES

A February, 1985 I/O letter asked about football handicapping programs. Several football statistical and prediction programs can be found in "BASIC Betting: the Microcomputer Edge," by James Jasper ($9.95, St. Martin's Press, NY). It covers baseball, basketball, football, and horseracing. It was intended for the Apple originally, but it should be possible to rewrite these programs for the Atari.

L. Allen Hummer
Fayetteville, PA

SEARS MONITOR GHOST

When **Antic** reviewed the Sears $349.99 **Proformance TV/Monitor** in our December 1985 Buyer's Guide, we wrote that it had a distracting color ghost when used as an Atari monitor. At the time, local Sears spokesmen assured us that the problem was a unique glitch in the unit we had borrowed for review.

As a result of monitoring the CompuServe Atari SIG, **Antic** has now discovered that the problem is far more widespread. One electronically oriented SIG member wrote that when he looked inside his Sears TV/Monitor he felt that its composite video mode circuitry (needed by the Atari) seemed like a quick add-on to what was essentially an RGB monitor intended for IBM-type computers. —ANTIC ED

i/o board

WRONG NUMBER

One of the BBS numbers **Antic** downloaded from the Boise Users' Group and reprinted unchanged in the February 1985 issue was incorrect. Please do NOT call the (601) 388-3940 number in Mississippi—it does not belong to a bulletin board. —ANTIC ED

JOYSTICK SPRAYPAINTER

I found "Spraypainter" (**Antic**, October, 1984) a little slow, so I converted it to ACTION! and installed an on/off routine with the joystick to make it more usable. Here it is:

```
BYTE y,s,i,y1,
  div=[25],
  ofs=[5]
CARD x,x1

PROC Init()
  Graphics(8+16)
  Setcolor(2,0,0) color=1
  Plot(100,100)
  x=100 y=100
RETURN

PROC Joystick()
  s=Stick(0)
  IF s<8 AND x<313
    THEN x=x+1 FI
  IF s>8 AND s<>15 AND x>6
    THEN x=x-1 FI
  IF (s/4)*4=s-1 AND y<183
    THEN y=y+1 FI
  IF (s/2)*2=s AND y>6
    THEN y=y-1 FI
  IF Strig(0)>0 THEN
    Plot(x,y) color=0
    Plot(x,y) color=1
    RETURN
  FI
  FOR i=1 to 4
   DO
    x1=x+Peek(53770)/div-ofs
    y1=y+Peek(53770)/div-ofs
    Plot(x1,y1)
   OD
RETURN

PROC Spray()
  Init()
  DO
    Joystick()
  OD
RETURN
```

William Bennett
San Antonio, TX

EXPANDED CPU? NO.

Is it possible to put a CPU expander bus into my 1200XL?

Greg Metallmos
Winnipeg, MB

We checked with Bill Wilkinson, who tells us that putting a bus expander on the 1200XL is out of the question for all but the most experienced electronics technician. Even if you could, it wouldn't be compatible with any other model without some very expensive conversion hardware and difficult-to-write software, says Bill, adding "Forget it."—ANTIC ED.

TWO-FACED FLOPPIES

Can you notch a single-sided disk and use the other side? If so, will it cause any harm?

Raymond Moody
Fort Ord, CA

1. Yes . . . Notching a disk is easy—you can just use a regular hole punch. To be sure of putting the notch in the right spot, hold an already-notched disk behind the disk you're punching.

2. Possibly . . . You run a slight risk in using the floppy's flip side. A disk drive's read/write head presses the bottom of a disk against a felt pad. Pieces of dirt or metal could get stuck in the pad and scratch the "A" side of your disk if you're recording on the second side. However, at ***Antic*** *we routinely use both sides of disks all the time and haven't lost any files yet. It's up to you if you're willing to take even a tiny risk with your disk data.*

—ANTIC ED

CHIPS, CHIPS, EVERYWHERE CHIPS

What can you tell me about the Western Design Center's OXI-CMOS W65SC802 CPU or related chips? It is supposedly a 16-bit processor compatible with existing 6502 applications. The chip is compatible, pin for pin, with the 6502 used in Ataris.

Mike Rutledge
El Segundo, CA

We checked with Charles Cherry, of Technical Support in Daly City, who supplied the following information and short history of the Atari 6502. —ANTIC ED

The Atari 400 and 800 use the 6502B, a faster version of the original 6502 microprocessor. The 600XL and 800XL use the 6502C, a substantially different chip that incorporates support functions that, in the days of the 6502A, were contained on separate chips. A further consolidation of support chips led to the 6510, which may be used in the new XE computers.

There are three other interesting chips in the 6502 family.

The 65C02 (a plug-in replacement in the Atari 400 and 800) offers the increased reliability, decreased power consumption and heat generation, and better heat immunity of CMOS. It also has new machine language instructions and addressing modes, which are supported by the MAC/65 assembler cartridge from O.S.S.

The other two chips are 8-bit and 16-bit processors based on the 6502. The 8-bit W65SC802 has new instructions and addressing modes. It appears to have the same new capabilities as the 65C02, and may work with MAC/65.

The 16-bit W65SC816 chip probably won't work with the Atari because of the pin arrangement.

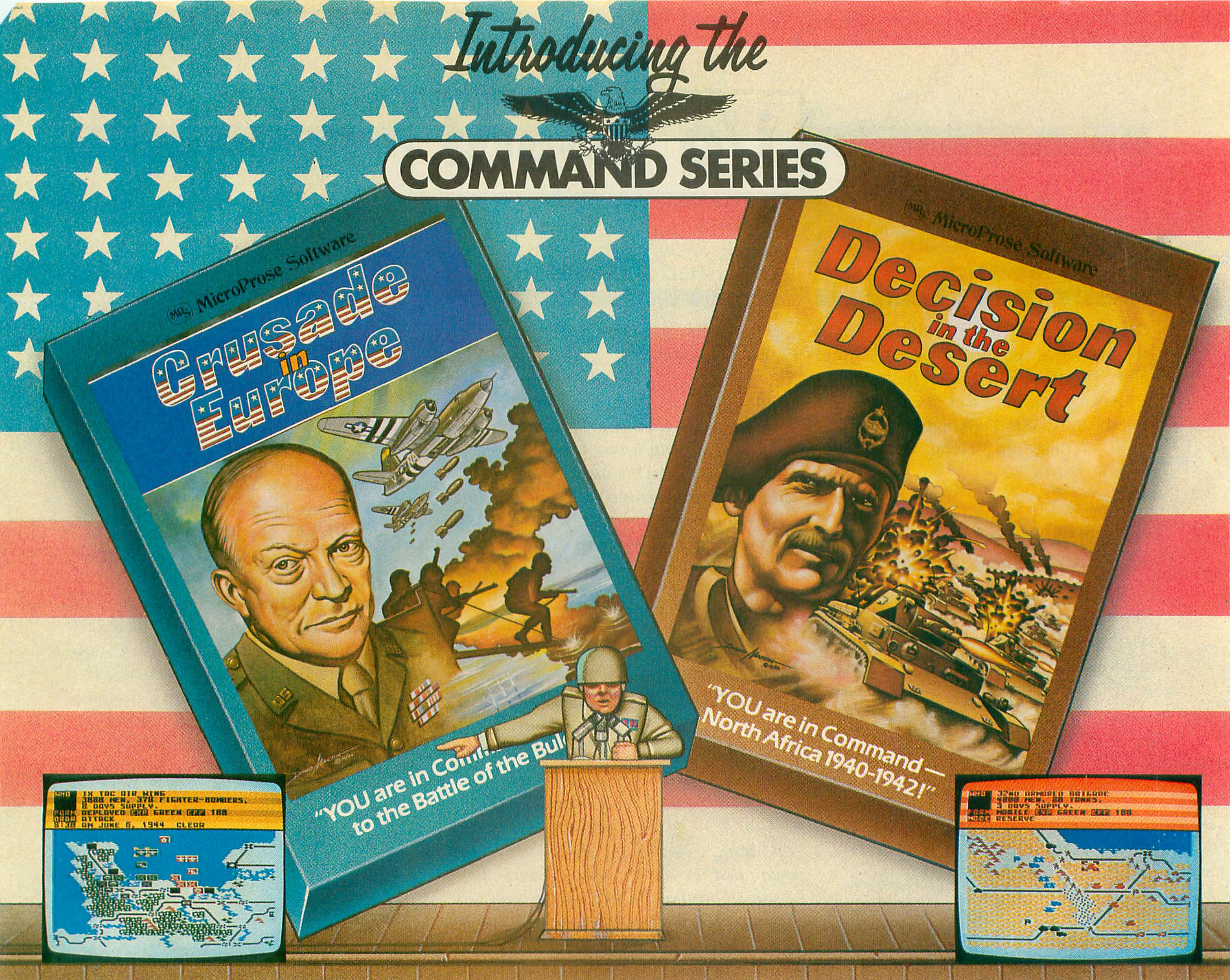
help!

ADVENT X-5 AGAIN

We have found that even with the missing line (8020 RUN) included, readers are having problems with ADVENT X-5. Take a close look at line 1005: the third inverse P in the second line is lower-case, and the thirteenth character in the second line (just before the inverse f) is a CTRL-B.
—ANTIC ED

PENCILS ON DISK

Antic omitted to put "Pencils" onto the March disk, as we had promised on the microscreens pages. So the nifty GTIA image by Gregg Tavares will appear on the disk for the next issue. —ANTIC ED



Introducing the
COMMAND SERIES
MicroProse Software
Crusade in Europe
MicroProse Software
Decision in the Desert
"YOU are in Command—
North Africa 1940-1942!"

TSCOPE AUTODIALER

Automatic log-on program

by CHARLES JACKSON, **Antic** Program Editor

A short, automatic log-on program for TSCOPE, the popular public domain telecommunications program. It will run on any Atari computer with a disk drive. Works with any TSCOPE compatible modem, including the Atari 1030 & 835.

TSCOPE, by Joe Miller, is a well-known public domain program. It enables owners of the Atari 1030 or 835 modem—or most modems that work with the Atari 850 interface—to upload and download either binary or ASCII files on the CompuServe SIG Atari.

(TSCOPE is available on the Antic 1030/835 Telecommunications Disk, PD025 in the Antic Catalog. —ANTIC ED)

TSCOPE Autodialer is a fast and foolproof way to log onto CompuServe automatically. You just boot your TSCOPE disk and sit back while TSCOPE Autodialer types in your CompuServe phone number, your User ID and your password.

When TSCOPE starts, it looks for a file named AUTODIAL.SYS which contains a simple set of log-on instructions. AUTODIAL.SYS is optional and doesn't come included with most versions of the TSCOPE program. You must create your own.

GETTING STARTED

TSCOPE Autodialer will create an AUTODIAL.SYS file for you. Type in Listing 1, checking it with TYPO II, and SAVE a copy to disk.

When you RUN the program, it will ask you for the phone number to dial, your access number (User ID), and your password. When you've entered this information, place your TSCOPE disk into the drive and press [START] to write the file. If you haven't already renamed your TSCOPE.OBJ file AUTORUN.SYS, the program will remind you to do so.

TSCOPE Autodialer creates an AUTODIAL.SYS file which might look something like:

```
555-1234
∧C]:98765.4321
]:SECRET.PASSWORD
```

The first line contains the phone number to be dialed. Hyphens, parentheses and blank spaces are ignored by TSCOPE.

The second line begins with ∧C—the code for [CONTROL] [C]. The right-bracket symbol "]" after the "C" stands for "wait". This tells the autodialer to wait for a prompt before continuing.

With our sample AUTODIAL.SYS file, TSCOPE would dial 555-1234 (ignoring the hyphen) and wait for a connection. Then your autodialer would issue a [CONTROL] [C] code and wait for the colon [:] at the end of the User ID: prompt. When the autodialer receives this colon, it enters your access number.

The colon on the last line of the AUTODIAL.SYS file tells the autodialer to wait for the next colon—the one at the end of the Password: prompt. It then enters your password.

continued on next page

MAKING CHANGES

There are many ways to modify your autodialer. If you wanted to automatically visit the ANTIC ONLINE service, for instance, you would add this line to your AUTODIAL.SYS file:

^]!GO ANTIC

This instruction tells the autodialer to wait for an exclamation point prompt, then type the GO ANTIC command.

PASSWORD PROTECTION

Though TSCOPE Autodialer is the quickest way to log-on to CompuServe, it lacks some security. Anyone who can load a disk and turn on a modem could gain access to your CompuServe account. So always keep your autodialing TSCOPE disk in a safe place. Listing on page 80.

In April, type GO ANTIC as soon as you log onto CompuServe. You'll be able to read **Antic's** immediate on-the-spot coverage of Atari news from the 1985 West Coast Computer Faire, which took place from March 30 to April 2.

ANTIC ONLINE will also give you a full preview of the stories and programs in the next **Antic** Magazine—the June **Computer Arts** issue.

You'll even see a major excerpt from the upcoming issue's featured article. This time it's a look at two breakthrough music products—an Atari MIDI controller that emulates a 16-track digital recording studio, and a real-time music generator that lets you improvise four-part compositions at the Atari keyboard.

Each month ANTIC ONLINE brings you the very latest Atari information long before it can appear in any magazine. From most areas there are no long distance charges for this service, so it does not cost you anything more than the standard CompuServe online time charge

The ANTIC ONLINE special bulletins may be downloaded for reprinting in newsletters of users groups affiliated with the Antic Worldwide Users Network. Officers of Atari users groups may write to **Antic** for details about WUN affiliation.

Imagine...

FOR COMMODORE 64
ATARI
APPLE IIe/c AND IBM PCjr

A WORD PROCESSING PROGRAM!
AN INFORMATION MANAGEMENT PROGRAM!
A TELECOMMUNICATIONS PROGRAM!

ALL THREE PROGRAMS, ON ONE DISKETTE!, FOR ONLY $49.95*!

"Quite simply the best! The highest rating possible... the package should be part of every (computer) library." — ANALOG COMPUTING

"Russ Wetmore has done an EXCELLENT job! The program is flexible, powerful and very easy to use. $49.95 buys a heck of a lot of program." — REVIEW BY ARTHUR LEYENBERGER

"Performance: ★ ★ ★ ★ (Excellent) Value: ★ ★ ★ ★ (Excellent)
This three-in-one package is a bargain . . . one of the finest values on the market."
— FAMILY COMPUTING

HOMETEXT WORD PROCESSOR.
HOMEFIND INFORMATION MANAGER.
HOMETERM TELECOMMUNICATIONS

Together they are **HomePak**: the three most important and most useful home computer applications in one integrated system — on one diskette!

The reviewers are unanimous: any one of these programs **alone** is well worth the price. So you're getting **three times the computing power**, with this exceptionally easy to use package:

- **all commands in simple English; no complex computer jargon, no obscure instructions**
- **all key commands are immediately available on the screen menu; additional commands can be called up for the more experienced user**
- **to help you, system status is displayed right on the screen**

And it's easy to use the three programs together. For example, in the "Merge" mode, you can take data stored in HOMEFIND and print letters and labels using **HOMETEXT**. Or, use **HOMETEXT** to write reports based on information you've called up via **HOMETERM**.



BATTERIES INCLUDED

30 Mural Street
Richmond Hill, Ontario
L4B 1B5 Canada
(416) 596-1405
Telex: 06-21-8290

17875 Sky Park North
Suite P, Irvine, California
USA 92714
Telex: 509-139

"The Energized Software Company!"

WRITE FOR A FULL COLOR BROCHURE

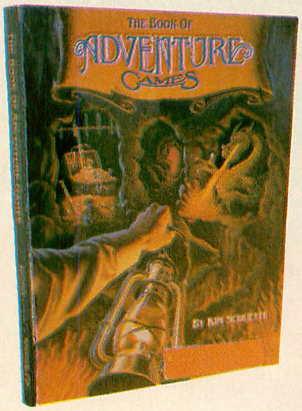
*MANUFACTURERS SUGGESTED U.S. LIST PRICE.

EACH COMPUTER SYSTEM MAY REQUIRE ACCESSORY DEVICES SUCH AS MODEMS, PRINTERS OR CARDS TO UTILIZE SPECIFIC FEATURES OF HOMEPAK. SEE YOUR DEALER FOR DETAILS.

ANTIC BOOKSHELF

Reviewed by CHARLES JACKSON and MICHAEL CIRAOLO

Book of Adventure Games
by Kim Schuette
$19.95
344 pages, paperbound



Cheating is expensive. In this case, it will cost you about $20 to obtain maps and cheat sheets for most existing adventure games.

Of course, it might well be worth $20 to avoid those sleepless hours, as you pull out your hair and wonder how to get past that bear or enter those massive doors in your favorite adventure game.

Whether you want to use such a cheat book is your business. But if you do, you'll find this an excellent guide.

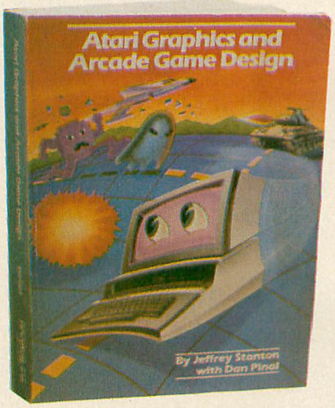
Maps and solutions reflect actual gaming experience and include appropriate editorial comments. The maps are well drawn and clearly presented. However, in our random sampling of game clues, some maps contained minor, but frustrating inaccuracies. Clues are provided as needed, in the form of numbered notes. These clues are in a separate section at the end of the book, so it *is* possible to just peek at that one answer you absolutely cannot figure out.

Each game also comes with publisher information, suggested retail price, description and brief review, necessary menus and character charts.

The Book of Adventure Games covers over 75 titles, most of which were designed for the Apple. But 42 are available for the Atari, including all Infocoms except the very latest, the Ultima series, the Adventure International catalog, Gruds in Space, Ulysses, Wizard and the Princess, and most other favorites.

This book is published by Arrays, Inc., 11223 South Hindry Avenue, Los Angeles, CA 90045.

Atari Graphics & Arcade Game Design
by Jeffrey Stanton with Dan Pinal
$16.95
479 pages paperbound



LINDA TAPSCOTT

Atari Graphics & Arcade Game Design was written for intermediate BASIC programmers ready to master the Atari at a higher level.

This is not a book for beginners who think a "Sprite" is something that goes well with a hot dog and a "redefined character" is a fellow who's had a spiritual experience.

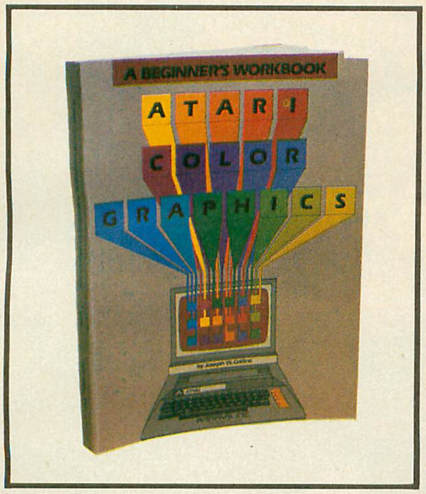
The early chapters deal with display lists, character set graphics and ANTIC and GTIA graphics modes. Several short BASIC program listings are included to illustrate key points in the text.

In a gentle introduction to Assembly Language, a BASIC version of a "Breakout" game is taken apart and its subroutines are explained. Ensuing chapters compare each subroutine to equivalent assembly language macros. By the time you're through, you should be a lot closer to designing and writing your own machine language arcade games.

Although the assembler listings are written in Synassembler, the book has a comparison table to help you translate the Synassembler code to Atari Assembler Editor, MAC/65, Atari Macro Assembler or Eastern House.

(This book is available by mail from the Antic Catalog bound into this issue of the magazine.)

Atari Color Graphics
by Joseph W. Collins
$12.95
202 pages paperbound



Atari Color Graphics: A Beginner's Workbook is a useful introduction to 14 Atari BASIC graphics modes. These include the three GTIA modes and two modes (Graphics 14 and Graphics 15) unique to XL computers.

If you're a beginning programmer, you'll want to keep your BASIC reference manual close at hand, since the workbook only describes BASIC graphics commands.

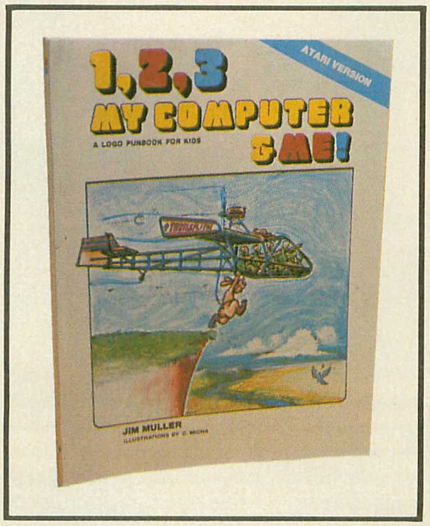
Each workbook chapter introduces a different style of computer graphics, including high, low and medium resolution modes; single and multicolor modes; the GTIA modes and three text modes.

The book contains many illustrations and dozens of short type-in programs that demonstrate key points in each chapter. New BASIC programmers ready to add interesting graphics routines to their programs should start with this book.

(This book is available by mail from the Antic Catalog bound into this issue of the magazine.)

Both graphics books reviewed here are published by Arrays, Inc./The Book Division, 11223 South Hindry Avenue, Los Angeles, CA 90045.

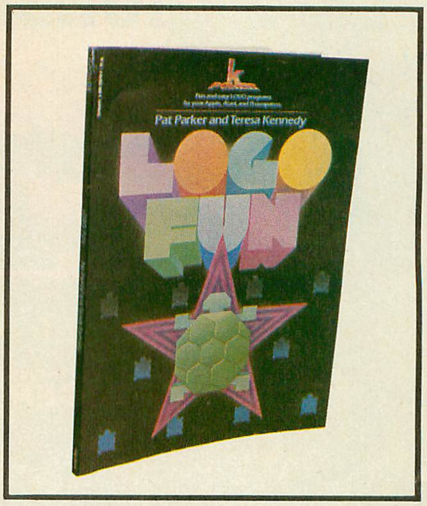
1, 2, 3, My Computer & Me! A Logo Funbook For Kids (Atari version)
by Jim Muller and the staff of the Young Peoples' Logo Association.
$12.95
111 pages paperbound



Here is one of the finest Logo workbooks available for children. Armed with this book, young people unfamiliar with Logo will quickly have turtles dancing on their screen. Later chapters explore recursion, music, writing and editing procedures and using the Logo shape editor.

Children will enjoy this lively and instructive book. It is filled with dozens of colorful and enjoyable Logo procedures to try. Parents and teachers will appreciate 1, 2, 3 because every lesson encourages children to use experimentation, imagination and intuition to solve programming puzzles.

Logo Fun
by Pat Parker and Teresa Kennedy.
$5.95
112 pages paperbound



This Logo tutorial simultaneously describes versions of the language for Atari, Texas Instruments, and two Apple variants. Consequently, you must be familiar with the Atari Logo user's guide before you read Logo Fun. Without this knowledge, debugging your Logo procedures soon becomes a frustrating nightmare.

Logo Fun contains a wide assortment of tiny procedures which draw attractive patterns on the screen. Several of these designs are presented in an eight-page color section in the middle of the book.

The authors invite you to use their book like an encyclopedia—to "flip back and forth, or check the index to find what you need." Unfortunately there is no index, and "flipping back and forth" soon becomes a time-consuming chore.

Both Logo books reviewed above are from Reston Publishing Company, 11480 Sunset Hills Road, Reston, VA 22090. (800) 336-0338.

LOOKING FOR STRATEGY GAMES FOR YOUR ATARI®?



YOU'VE JUST FOUND SIX OF THE BEST.

OBJECTIVE: KURSK™ is the grand-tactical simulation of the southern half of the Battle of Kursk – the largest tank battle of World War II. Consider its scope: 12 German divisions and 9 Soviet corps – a sum total of more than 4000 tanks! It is the first computer game ever to resolve such a massive battle down to every tank, every gun, every infantry squad! $39.95.

REFORGER '88™. Once every year, NATO embarks on "Reforger," a military exercise that tests its ability to transport American reinforcements from the U.S. to Frankfurt during a simulated Soviet strike of West Germany. Now, REFORGER '88 makes the simulation available to every ardent wargamer. The Fulda Gap is chosen as the focal point of the Russian attack in this grand-tactical wargame. $59.95.

All games are on 48K disk except for 50 MISSION CRUSH (40K disk).
ATARI® is a registered trademark of Atari, Inc.

COMPUTER AMBUSH™ is a gut-wrenching simulation of man-to-man combat in the middle of a half-ruined French town during World War II. You play a squad sergeant (U.S. or German) in command of nine other infantrymen. Each man has a name, individual combat skills, even a personal background! The fighting is so fast, so real and intense, you'll experience the sweat and death of war! $59.95.

COMPUTER BASEBALL™. Voted "1982 BEST COMPUTER SPORTS GAME" by Electronic Games Magazine, Computer Baseball™ lets you manage any NL or AL team of past and present. All the options of a real manager are at your disposal. You can even make up your own imaginary teams! $39.95.

50 MISSION CRUSH™. As the pilot of a World War II B-17 bomber in this role-playing game, can you survive 50 dangerous but exciting raids over France and Germany to earn the crushed cap of a true veteran? Find out as you travel back to 1942 as part of the 8th Air Force Bomber Group. $39.95.

GALACTIC ADVENTURES™, a science-fiction role-playing, tactical combat, adventure game, will transport you literally out of this world – to a spaceport of a strange planet. You must acquire combat experience and advanced skills before you can fly off to different worlds in search of treasures and to do battle against alien monsters. You can even create your very own adventures. $59.95.

Look for these games at your local computer/software or game store today!

SSI

STRATEGIC SIMULATIONS INC

If there are no convenient stores near you, VISA & M/C holders can order direct by calling **800-227-1617, ext. 335 toll free.** (800-772-3545, ext. 335 in California.) To order by mail, send your check to: Strategic Simulations Inc, 883 Stierlin Road, Bldg. A-200, Mountain View, CA 94043. Please include $2.00 for shipping & handling. (California residents, add 6.5% sales tax.) **All SSI games carry a 14-day "satisfaction or your money back" guarantee.**

FOR ALL SSI GAMES, WRITE FOR YOUR FREE COPY OF OUR COLOR CATALOG.

the Hitchhiker's Guide to the Galaxy



Reviewed by JACK POWELL and MICHAEL CIRAOLO

We know how to get the Babel Fish. But don't ask us. We won't tell you. And Don't Panic, the clue is right there in the game.

We're talking about Infocom's new text adventure, **The Hitchhiker's Guide to the Galaxy**, based upon the first of that insanely funny series of books by British author and ex-bodyguard Douglas Adams. If you haven't read the book, please do. It will definitely help you in the game.

For those culturally deprived members of our audience, the game generally follows the cult-classic book although Adams did write extensive (and very funny) new material for the adventure.

The excitement opens as you awake to a hangover in your bed in Cottington, England. Playing the part of Arthur Dent, hapless earthling, you must quickly come to terms with existence . . .

Your house is about to be demolished to make way for a highway bypass. No matter, really. The earth is about to be destroyed by a Vogon Constructor Fleet to make way for a hyperspace bypass.

But wait! There's more! Infocom takes YOU, the feckless adventurer, to worlds beyond imagination: "Welfare planets ruled by dry-cleaning establishments, where even the most basic of human necessities are provided a day late and with too much starch." And so on.

Face it—this is not your run-of-the-mill text adventure. If you're going to survive, you'll need your trusty Hitchhikers Guide (built into the game) and a towel! Be warned: Despite its "standard level" rating, this is the most challenging game we've seen from Infocom. (Have YOU gotten the Babel Fish or bested the Ravenous Bug-Blatter Beast of Traal?)

The puzzles are tough, but they follow a certain capricious, twisted internal logic. As we played, we encountered repeated dead ends. When we finally discovered the answers, we found the solution was logical and often accompanied by previous clues.

In fact, if you stumble around enough in certain problem areas, the computer will eventually throw in a hint.

The best way to understand this British whimsy is to read and enjoy Adams' books or possibly the works of Lewis Carroll.

This extraordinary game is the result of an unusual partnership. Adams, who is a long-time fan of Infocom games, approached the company with the possibility of doing a game based on his book. He teamed up with Steven Meretzky, the award-winning author of Infocom's **Sorcerer** and **Planetfall**.

The result is a step forward from Infocom's safe, established approach to game design. It is a break from the tradition of event-specific mysteries and plotless underground dungeons. The style of writing is distinct and tangible—really the *first* stylistic departure since the classic Zork trilogy.

Tips for novices: play the game with a grizzled Infocom adventurer OR a crazed Hitchhikers fan.

And now, we now have a confession to make. We had planned on getting this review into print at least a month ago, but we hoped to finish the game first. Alas, we simply haven't been able to get past the Screening Door. So, if anyone out there has a clue. . .

This text adventure is available from Infocom, Inc., 55 Wheeler Street, Cambridge, MA 02138, phone (617) 492–1031. $34.95, 48K-disk.

by CHARLES CHERRY

GEM of ATARI

More than pretty icons!

When Jack Tramiel announced that the new 16-bit STs would use the GEM operating environment, he joined Atari to one of the most innovative lines of research in computer history.

During the 1970s a group of digital visionaries gathered at Xerox's Palo Alto Research Center (PARC) to explore how computers should relate to people. They wanted to teach "people literacy" to computers instead of computer literacy to people.

It is already hard to remember how difficult it used to be to operate computers. You literally needed a computer science degree to use them. But video screens and electronic keyboards replaced punch cards and teletypewriters in the '70s. And in those new video terminals, the PARC dreamers saw the future. . . A video screen could show anything, and a picture could replace a thousand words.

IDEA PICTURES

The icon was born, a picture of an idea. Like international traffic signs, an icon can communicate more quickly and more vividly than words. A file cabinet represents a database, a piece of paper stands for a word processor, a disk means DOS. Since icons can be small and simple, many can be put on the screen without confusion. You can see all of the available options

simultaneously. All you need to do is select among them.

But how do you select an icon? The gang at PARC tried everything, keyboards, touch tablets, light pens, joysticks and finally a mouse. The mouse was their choice—simple, natural and intuitive.

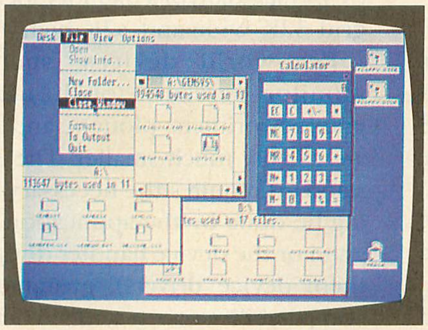
There was one more element to invent, a visual metaphor for the way you use a program. As you do various things in a program the entire screen keeps changing. For instance, if you want to change the skill level of the game you are playing, the playfield disappears and the option screen appears.

But the PARC researchers thought this was wrong. You should not have to jump around in a program, the program should come to you. Hence the idea of windows. A portion of the playfield would open up and reveal enough information for you to make your choice, while the rest of the game is still visible behind the choices.

PARC TO MAC

These visions led Xerox to build the dream machine called Star. It was wonderful and elegant and over $20,000—much too expensive for the 1981 market.

Steve Jobs of Apple got access to look around inside PARC and a year later Apple's Lisa came out. Priced around $10,000, it did not sell very well either. But it attracted lots of attention. Then Apple tried again with the Macintosh, which was originally priced at $2,495. The PARC vision was finally within reach of the general public.



GEM desktop display for the New Atari ST

In the above three computers, both software and hardware systems were custom designed as one complex unit. But underneath the spectacular screens were ordinary computers. The hardware had the same input and output requirements, the same memory management problems. The PARC graphics environment was simply an overlay—which in theory could work with any operating system on any computer.

The challenge of creating a single graphics environment overlay which would be compatible with many different computers was taken up by Digital Research, Inc. of Monterey,

continued on next page

California. DRI had developed the first microcomputer operating system, CP/M. Now they produced the Graphics Environment Manager, GEM.

Although it may well run on other operating systems in the future, GEM is currently available for IBM PC-DOS, and for the closely related MS-DOS and Concurrent DOS.

GEM MEETS ATARI

GEM has now also been chosen for the upcoming Atari ST computers. It will work with the new TOS operating system, which is a close relative of DRI's CP/M.

In the Atari STs, the GEM overlay, the TOS operating system, and the device drivers including hard disk, floppy disk, Centronics parallel, and RS232C serial are all to be contained in 192K of ROM . That means when you turn on your ST it is ready to work immediately and none of your RAM has disappeared.

GEM does more than make computers easy to understand and use, with flashy icons and drop-down menus. It also supports multi-tasking. That means that you can run several programs at the same time and easily pass information between them. It is what we all thought computers could do before we got one.

GEM's powerful graphic capabilities are available to application software, so terrific drawing programs and spectacular games should be the order of the day. In the multiple windows you could run your word processor and your spreadsheet *at the same time.* Programs that we never dreamed possible will be!

INSIDE GEM

GEM works by setting up an imaginary all-purpose input-output graphics device, called a Virtual Device Interface (VDI). All graphics I/O is sent through it. This will sound familiar to programmers who have used the Atari CIO. The GEM VDI and IBM both follow the emerging American National Standards Institute (ANSI) standard of a memory location grid 32K wide and 32K high.

Real-world devices, such as monitor screens, touch tablets, graphics printers, plotters, and mice are usually much smaller. The ANSI standard requires the necessary scaling be done by the application program or the device handler (they call them device drivers), guaranteeing compatability over a wide variety of actual devices. GEM calls this the Normalized Device Coordinates mode.

GEM supports another VDI mode called Raster Coordinates (RC). This allows you to map the actual device coordinates over a portion of the VDI. It was designed with monitor screens in mind and permits addressing pixels directly, just like bit-mapped graphics. RC allows multiple screens to be created within the 32K by 32K VDI grid. You can then switch between them.

The GEM VDI supports over 50 functions. These are like the XIO functions and the AUX1 and AUX2 bytes in the Atari CIO. They handle setup of devices with defaults, graphic primitives like lines, polygons, ellipses, arcs, and others. They control color registers, line style, character fonts and cursor forms. They also support bit block transfers (which perform logic operations on bytes before moving them) and access to special device capabilities.

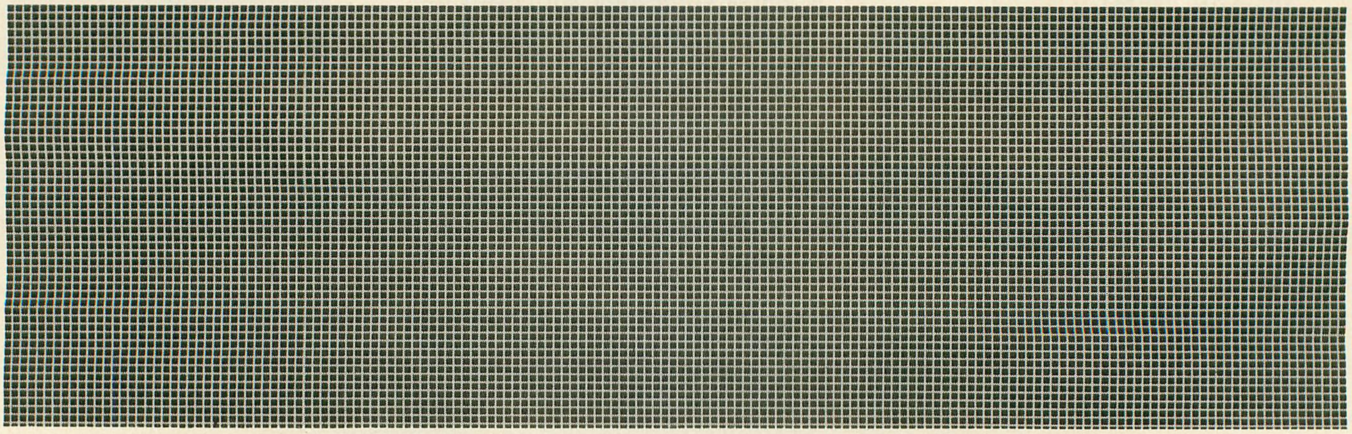
BUILT-IN LIBRARY

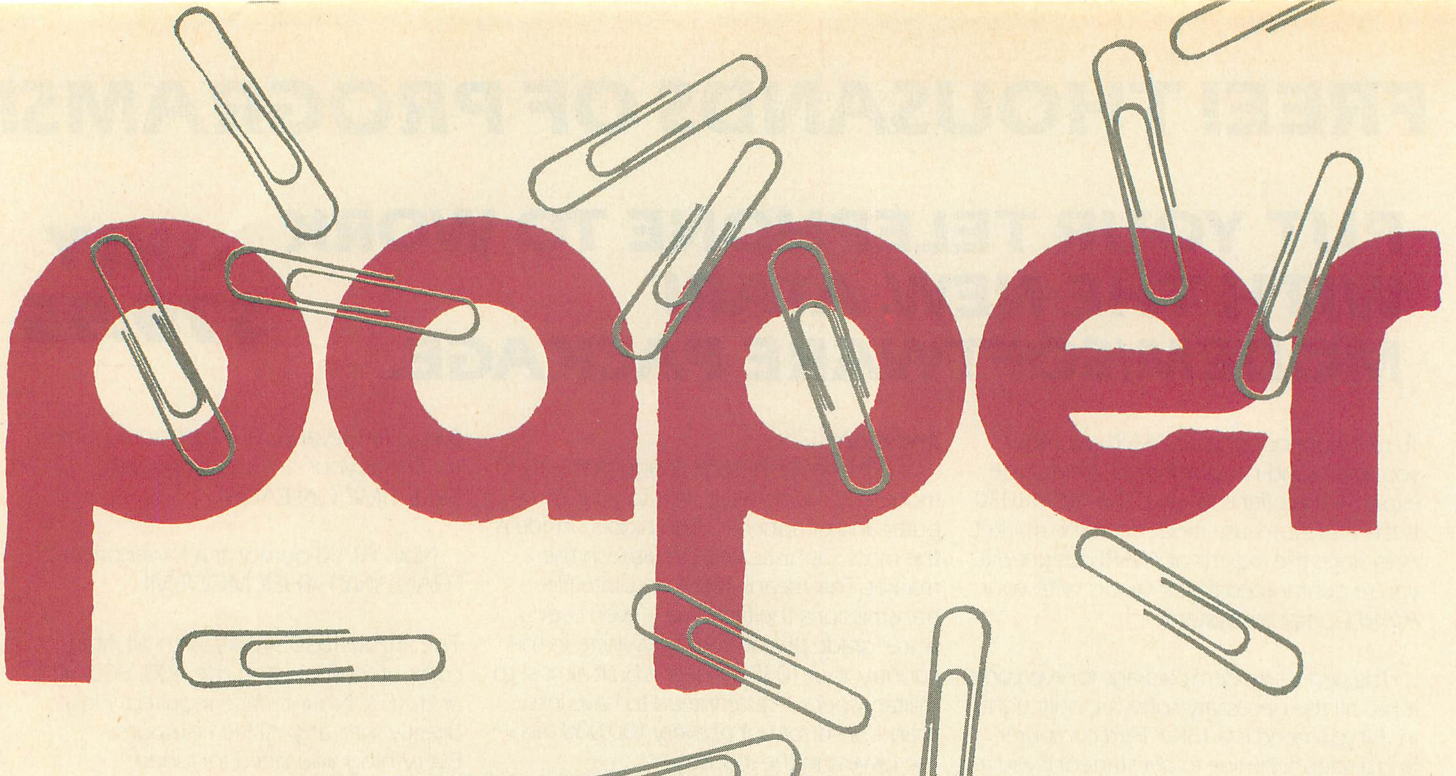
GEM includes a package of subroutine libraries in the Application Environment Services (AES). These libraries handle the program's interface with GEM, taking care of icon manipulation, drop-down menus, windowing, information transfer between applications, and a host of other details. This makes the GEM magic much easier to program. Digital Research also has just finished a GEM Programmer's Toolkit manual to aid in commercial program development.

Because GEM runs on many machines, most notably the IBM, programs are easily moved between operating systems. Consequently, it is believed that high quality IBM programs running under GEM will soon be available on the Atari. But this can work the other way, too. Atari programmers will be able to sell their creations in the IBM market. Finally, Atari owners will get some practical large-scale business programs and IBM owners will get some decent games.

The user interface in computers has come a long way in a very short time. We Atari users have had one of the best all along. GEM will be another giant leap forward.

Charles Cherry is a theatre technician who uses his Atari to generate images for industrial slide shows, as a moving message center, as a teleprompter and as a business machine.





by MICHAEL CIRAOLO and NAT FRIEDLAND
of the **Antic** Staff

In case you weren't aware of this, it's a matter of honor at **Antic** to use only Atari computers in our office. And since we are a publication, word processing software is used around here a lot.

We've noticed there are two schools of thought about word processing software for the Atari. Atari users who haven't had experience with other makes of computer are reasonably satisfied. But other Atari users who've had some exposure to machines with more of a "business computer" image unfortunately know better. . .

Antic had been using LJK's **Letter Perfect** as our in-house word processor—although without any great enthusiasm for it.

We'd found Letter Perfect to be rather more powerful and fuller-featured than **AtariWriter** or **Text Wizard**, the only other established "serious" Atari WP software.

Probably just as important to us was that only Letter Perfect would work with the 80-column cards we had at a couple of workstations. This advantage tended to make up for the program's files requiring a tedious conversion process anytime we needed to transfer in or out of standard Atari DOS.

ENTER PAPERCLIP

But the day our beta test copy of **PaperClip** arrived from Batteries Included in Toronto, the **Antic** Editorial Department cheerfully retired our Letter Perfect.

Every once in a while, a piece of software or hardware shows up here that is so clearly superior in its category to anything else available for the Atari that **Antic** starts using it in-house immediately. The previous example of this was "DISKIO" (January, 1985) that at once began replacing DUP.SYS on our program disks.

Simply, PaperClip is by far the best word processor ever available for the Atari. It boasts a line-up of advanced features that would be hard to match on even the biggest-name word processing software costing $300 or more.

PaperClip makes your Atari the word processing equal of just about any computer on the market. Yet it is not hard to learn and sells for only $59.95.

Some nuts and bolts information: PaperClip runs on all Atari computers with 48K. It comes on a disk that you can back up. But the program is protected by a special key that plugs into joystick port 2.

A few technical notes: As we've previously reported, it is based on the ACTION! editor and written in 100% machine language, so it's *fast*. (It's really quite different from the PaperClip version written for the Commodore 64.) The preliminary draft of the manual that we've got is pretty clearly written. And the screen gives you big, sharp letters with true descenders, because the program uses ANTIC Mode 3 and redefined characters.

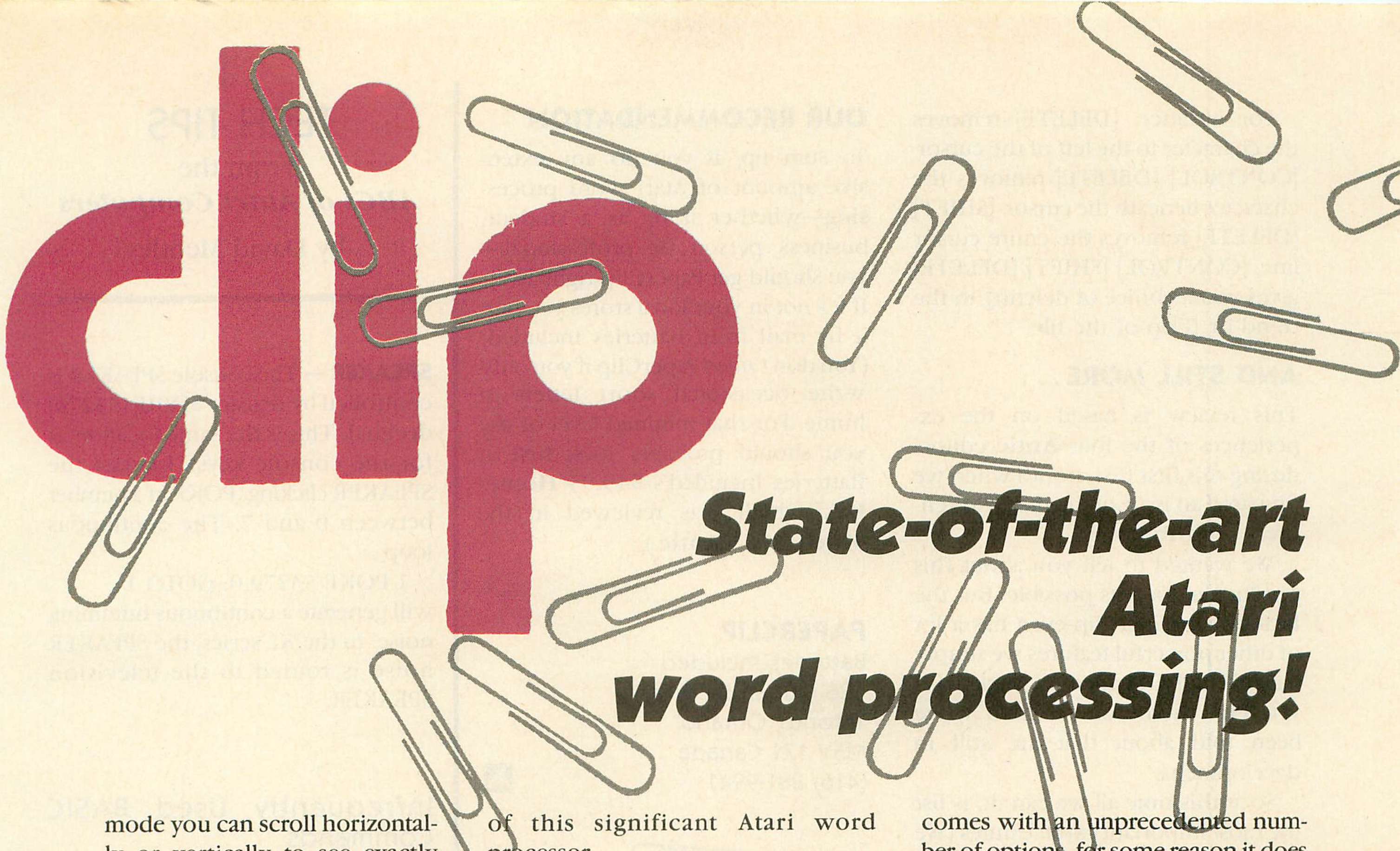
UNIQUE FEATURES

To justify our enthusiasm for PaperClip, here are some of its most distinctive features:

- DUAL WINDOWS—You can display two text files onscreen at the same time. And you can easily move blocks of text between file windows.

- ONE-KEY MACROS—You can easily set up for one keytouch while pressing [START] to type in an entire word, phrase or paragraph that you regularly use in your writing.

- PREVIEW MODE—PaperClip is easier to use in 40-column screen format than any word processor we've ever seen. In the preview



State-of-the-art Atari word processing!

mode you can scroll horizontally or vertically to see exactly how your words will fit on the page. Even in the normal mode, a diamond mark at the end of each line shows exactly where the word wrap is. (And PaperClip will support Batteries Included forthcoming plug-in 80-column card, which is due later this spring.)

- HIGH-POWER COMMANDS—There's actually no flipping between mode menus. Saving and loading files, disk formatting, editing, text entry, disk directories and help menus are all available from the same screen with the touch of very few keys. Some really unique and useful commands include automatic save, transposing letters or words, or converting capital letters and lower-case letters back and forth.

MINOR QUIBBLES

It is not the normal thing for a computer magazine to be able to review beta test software prior to its finalization for market. However, Batteries Included unconditionally agreed to let us rush a review of our beta copy of this significant Atari word processor.

The **Antic** editors did find some minor problems with PaperClip. But Batteries Included promised us that most of the bugs had already been fixed in the final version of the program that goes on sale in April.

Our biggest concern was the size of the memory buffer. The latest version we worked with had only enough free memory to handle a single-spaced document slightly over six pages long (or 12-1/2 pages double-spaced).

Batteries Included said the final version would hold files of about 20 pages double-spaced. The buffer in XL models will contain about 28K memory, 24K in the 800 model.

To set the print format commands for boldface, underline and italics, you must specify whether it is the beginning or the end of the formatted section. We found this cumbersome, especially when so many of the other commands are so convenient.

There are still a few things that Letter Perfect does which we wish PaperClip would also do. For example PaperClip does not have a command deleting an entire word, foward or backward.

Although this word processor comes with an unprecedented number of options, for some reason it does not let you turn off the keyclick in the 800 models, which have no independent volume control.

EASY POWER

Of course, in PaperClip you will also find all the standard features you'd expect from a competitive word processor today. There's global search and replace, underlining, italics, boldface, headers and footers, onscreen help files, pitch control, page length setting, nearly 30 different printer drivers plus a configuration menu, and on and on . . .

Yet for all the power it offers, PaperClip is surprisingly easy to learn. This is unusual, because the more powerful editors are usually harder to master. But PaperClip is virtually as easy to use as **Bank Street Writer** so there is no reason why it shouldn't be your first word processor.

Many of the editing functions are accomplished by holding down the [CONTROL] and [SHIFT] keys together plus a third key. With very little practice, this becomes second nature. And it also makes for an efficient command structure.

continued on next page

For instance, [DELETE] removes the character to the left of the cursor, [CONTROL] [DELETE] removes the character beneath the cursor. [SHIFT] [DELETE] removes the entire cursor line. [CONTROL] [SHIFT] [DELETE] gives you a choice of deleting to the [E]nd or [T]op of the file.

AND STILL MORE . . .

This review is based on the experiences of the four **Antic** editors during this first first month when we prepared an issue of the magazine entirely with PaperClip.

We wanted to tell you about this product as soon as possible. But the fact is that PaperClip even has a lot of other powerful features we simply haven't had a chance to work with yet. Plus there's one or two we've been told about that are still in development.

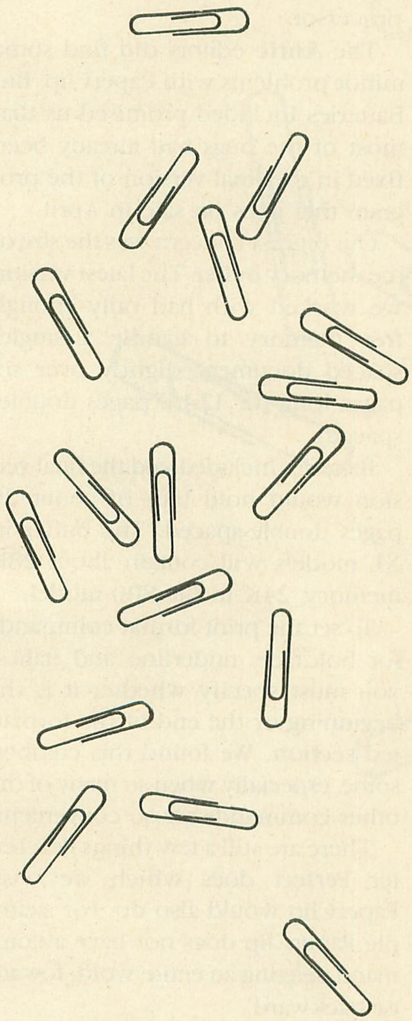
So at this time all we can do is list the most important extra features (We don't even have room for *all* of them) and promise to cover these extras in a later article or articles. . .

- TWO COLUMN PRINTOUT
- BUILT-IN MATH CALCULATOR
- MAILMERGE WITH **SYNFILE+**—Both programs are by the same authors, Steve Ahlstrom and Dan Moore, although SYNFILE+ was written in FORTH.
- MULTIPLE DISK FILE GLOBAL SEARCH—Up to 6 simultaneous search and replace operations throughout all linked disk files in as many as 4 separate drives. Truly amazing.
- ATARIWRITER-PAPERCLIP FILE CONVERSION—**Antic** Contributing Editor Jerry White is writing this one.
- MIXED TEXT/GRAPHICS SCREEN DUMP—This integrated screen dump will enable you to mix text and high-resolution Atari graphics (modes 7.5 and 8) on a single printed page. It's compatible with Micro Illustrator and most other graphics software files.

OUR RECOMMENDATION

To sum up, if you do any extensive amount of Atari word processing—whether it be as a student, business person or professional—you should get PaperClip right away. If it's not in your local stores yet, buy it by mail from Batteries Included. (You don't *need* PaperClip if you only write occasional short letters at home. For that minimal level of use you should probably look first at Batteries Included's $49.95 **Home-Pak** which was reviewed in the March, 1985 **Antic**.)

PAPERCLIP
Batteries Included
186 Queen St. West
Toronto, Ontario
M5V 1Z1 Canada
(416) 881-9941



TECH TIPS

From the
ABCs of Atari Computers
by David Mentley

SPEAKER — The console SPEAKER is controlled by register $D01F (53279) decimal. This is the same location as for the console keys. To start the SPEAKER clicking, POKE in a number between 0 and 7. The continuous loop

```
1 POKE 53279,0: GOTO 1
```

will generate a continuous humming noise. In the XL series, the SPEAKER noise is routed to the television SPEAKER.

Infrequently Used BASIC Commands

**STR$** — In BASIC, the STR$ command converts a number or numeric variable to a STRing. You will also need a string variable name to place the string into if you want to use it somewhere else. ONE23$ = STR$(123) will assign the string "123" to the string variable ONE23$.

VAL — In BASIC, VAL performs the opposite function as STR$. VAL converts a string which is made of numeric characters to a numerical variable or value. X=VAL(Y$) will assign the value of 123 to the variable X if Y$ were a string called "123". If Y$ is an alphabetic character, an error will result.

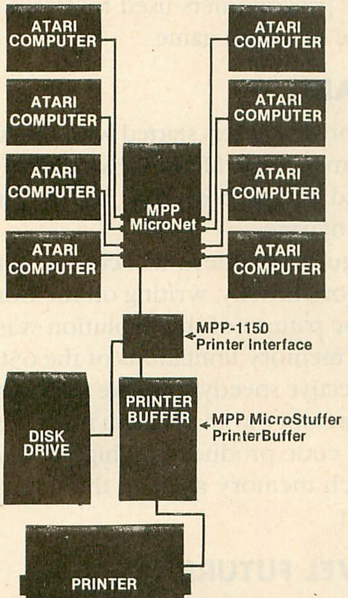
From *ABCs of Atari Computers* by David Mentley (available through the Antic Catalog in this issue). Reprinted by permission of Datamost, Inc.

MicroNet™

ONLY
$199.55



- Network up to 8 ATARI computers
- Allow sharing of Printers, Disk Drives, and Modems
- Works with all ATARI computers
- Provide enormous saving for school systems and computer labs
- Comes with modified ATARI DOS 2.5 to support busy disk retry



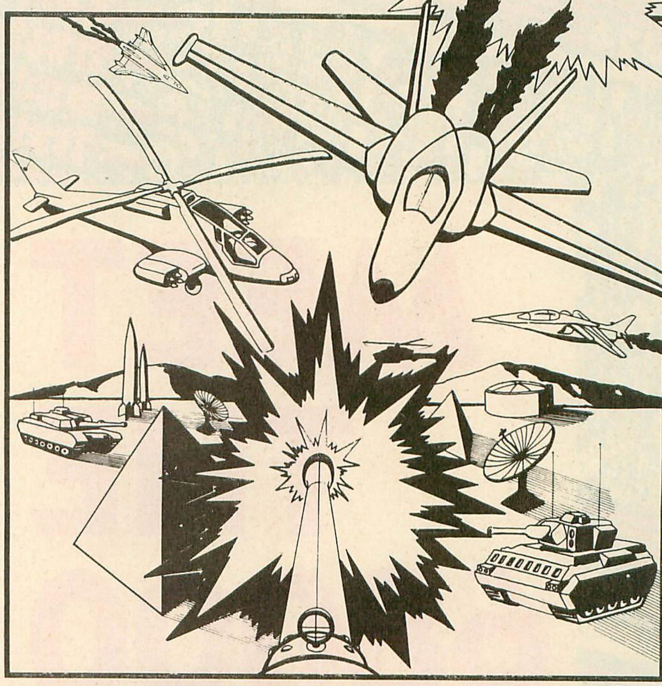
MICROBITS PERIPHERAL PRODUCTS

3615 Pacific Blvd., SW / Albany, OR 97321
ORDERS: 1-800-624-7532
CUSTOMER SERVICE: 1-503-967-9075
TWX: 910-997-6280

Works with any ATARI Computer with 48K and Disk Drive.

ASSAULT FORCE™ 3D

ONLY $39.95



Exciting new "real" 3D Graphics on 48K Disk. Includes Super Charger Power Booster cartridge for fast realistic 3 dimensional graphics. Weave your way through a maze of pyramids and radar dishes as you fight against an onslaught of jets and helicopters. But, remember to keep an eye out for more fuel in this fast paced game.

MicroPort™-XL

For Advanced Experimenters

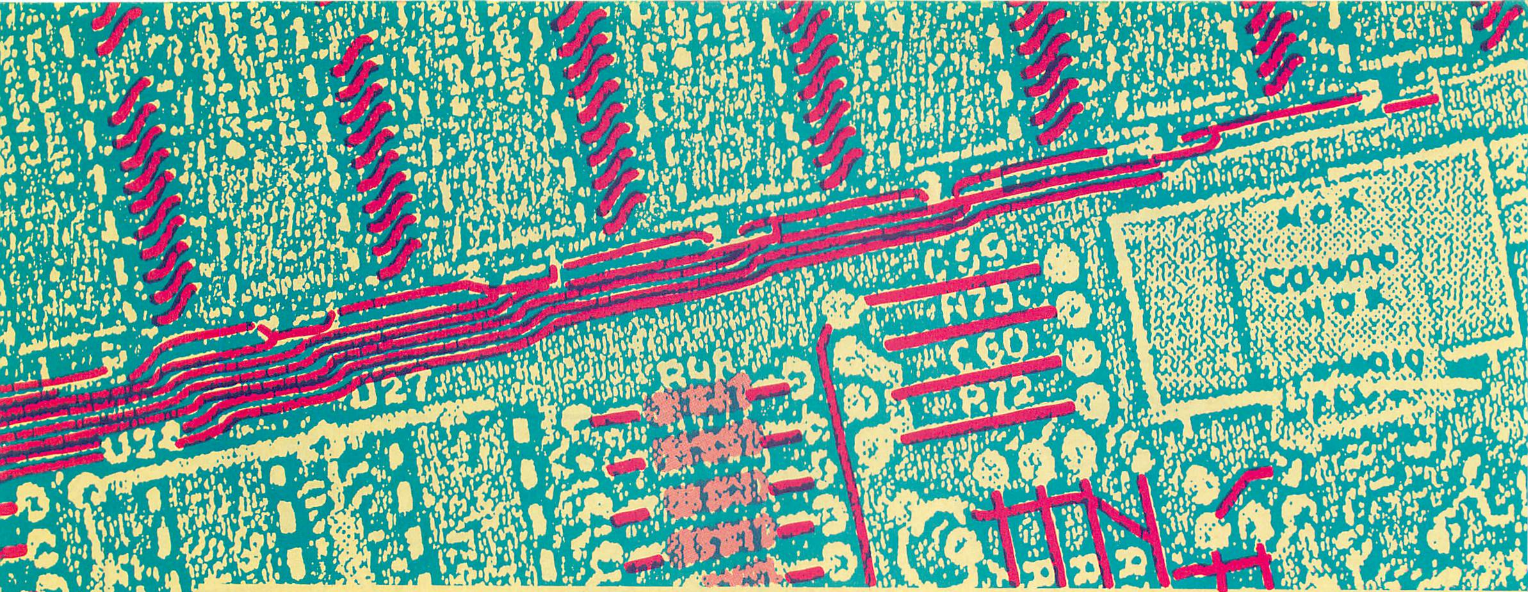
Parallel Buss Board-$49.99

This unit is designed as an experimenter's PIA board for the Atari XL computers.

All vital parts are common chips that are socketed for easy replacement.
Powered by an external source that also gives a 5 volt lead for experiments.
Switch selectable to blocks $D5, $D6, and $D7.
Provides 16 I/O lines and 4 handshake lines that can double as I/O's.
Based on the 6520/6821 PIA chip.
Will do interrupts.
Uses the parallel buss.
Over 10 sq. in. of drilled area for wire wrap prototypes.

Suggested uses include:

- VCR-Video Disk Controller
- BSR Home controller
- EPROM Programmer
- Monitor and Control Home Environment
- Music Synthesizer Control



MEET THE 68000

Will the ST 16-bit chip kill BASIC and assembler?

by JACK POWELL
Antic Technical Editor

There's a new kid on the block. He's bigger, brighter, more sophisticated, and a whole lot faster. His name is 68000 and he speaks in new tongues. There are those who say he could be death to assembly language and BASIC.

The new Atari ST computers will contain the Motorola MC68000 microprocessor chip. Introduced in 1979, it was the first 16-bit microprocessor to have 32-bit internal architecture with 16-byte, non-segmented direct memory addressing, which means you can potentially access 16,777,216 bytes. That's over 5,500 pages of single-spaced typing. For programmers used to Atari's old 6502 chip, it's a whole new ball game.

BASIC BABEL

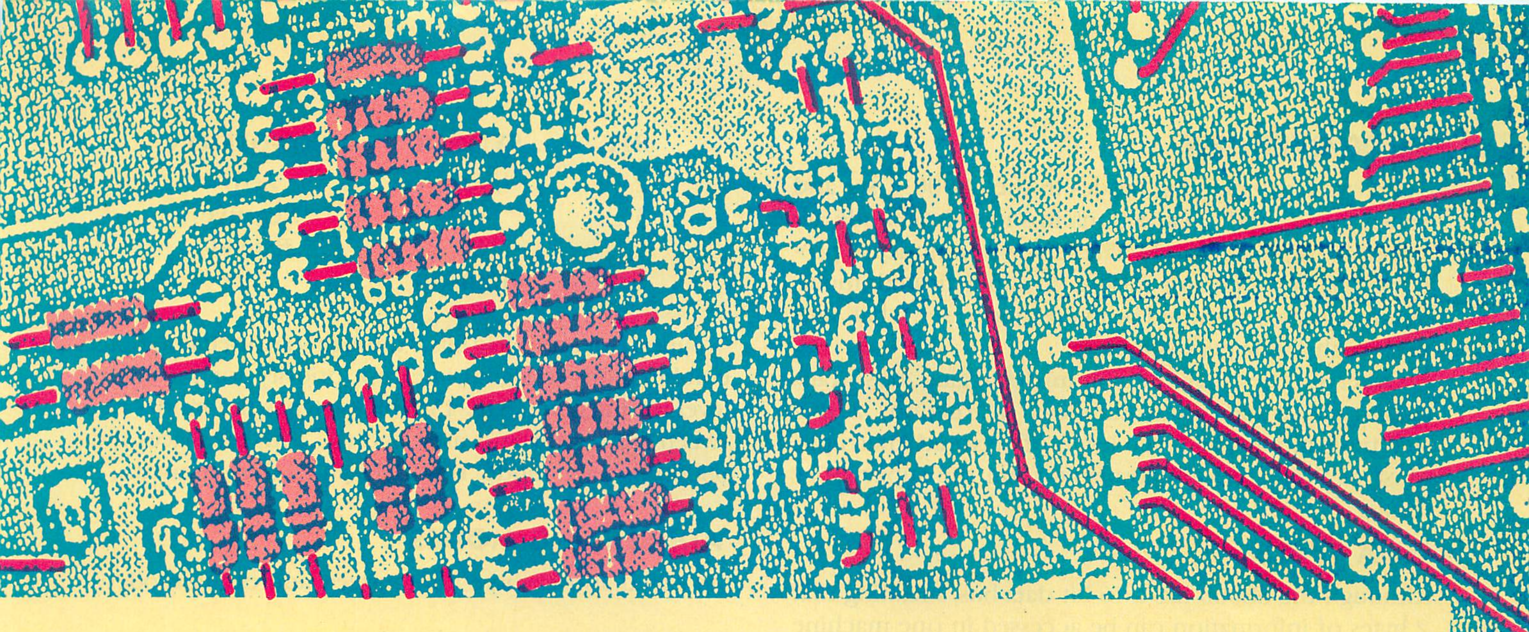
Most Atari programmers started with the BASIC that came with their machines. Many grew tired of its limitations, and evolved to the powerful, highly precise world of machine language.

Some argue this pattern will change with the new STs. Jeffrey Gortatowsky, writing on the Compuserve Atari SIG, says the pattern of this evolution was caused by the design and memory limitations of the 6502. If you want to write effective speedy software for a computer that has only 64K memory available, you must use assembly language. The code produced by high-level language takes up too much memory and, on the 6502, is usually not fast enough.

HIGH-LEVEL FUTURE

Gortatowsky then argues that, in the 68000, we will see the beginnings of change. The chip architecture is so flexible and fast, and the addressable memory so vast, that high-level language will become the practical choice of software manufacturers because time-consuming, costly assembly language will no longer be necessary. In addition, private owners may find that any form of BASIC which takes advantage of this chip's power is going to be cumbersome and difficult.

This theory is echoed in the fine reference book, **68000 Assembly Language Programming**, by Kane, Hawkins



& Leventhal (Osborne/McGraw-Hill, Berkeley, CA, 1981). 'We expect the future will favor high-level languages."

As chips and hardware become cheaper and more powerful, the theory goes, the cost of programming labor is rising. This helps explain the shift to efficient high-level languages for program development. These languages can also permit standardization of software development between machines.

A case can also be made that successful research in high-level structured languages is lagging behind hardware breakthroughs—which means that dramatic improvements in features such as ease of use are somewhat overdue and might be expected to happen reasonably soon.

Until then, it is not surprising that professional program developers are specifically being told by Atari to use C or Pascal on the STs.

There are two forms of C currently available for the Atari. **Deep Blue C**, which can be ordered from the Antic catalog, and **C/65**which you can get from Optimized Systems Software. Because of the Atari computers' current memory limitations, neither of these C versions are full implementations of the language. They should, however, serve as good training for those who wish to learn C.

The new ACTION! language, also from O.S.S. is a cross between C and Pascal. It's fun to program with and would also be an excellent training ground for those interested in learning structured programming.

In the computer world outside Atari, C is rapidly becoming more and more popular among home programmers. Does this mean BASIC is on the ropes? Not really.

A choice of BASIC or Logo—in versions designed by Digital Research who created the GEM operating environment used by the ST—will be bundled with the first ST machines. And there is such an established, wide base of BASIC programmers, it's doubtful the language will disappear.

New BASICs are likely to appear which will take advantage of extra memory and all sorts of easily-used new commands will be added. The original structure of the language, however, probably will remain the same.

MEMORY LIMITATIONS

And what about assembly language? We think memory-efficient AL will be as necessary and popular on the STs as on the earlier Ataris. Unlimited memory is not really available. The 68000 microprocessor may be *capable* of addressing 16 mega-bytes of memory, but, of the two announced ST models, the 130ST ($399) will contain only 128K of RAM and the top-line 520ST ($599) will have 512K.

On the IBM PC the highly touted Symphony, written in a high-level language, requires 340K just for starters. Framework, also written in a high-level language, will operate (barely) in 256K on the IBM PC— but its tutorial disk will not!

Atari says the new STs have non-expandable memory. We suspect it won't be long before some imaginative third-party manufacturer figures out a way to plug more memory into the "non-expandable" STs. But meanwhile if a software developer wants to market a program that will run on both ST models, assembly language may be the only viable solution.

HACKER'S PLEASURE

A consideration not taken into account in all this is the pure satisfaction that assembly language programmers get from programming at the nitty gritty level. We're no longer talking about the practical, economic business approach, but the home hacker who wants to roll up his sleeves and get to know every board, chip and register in that machine. He does not want to be limited by someone else's idea of what the processor can do. If he wants a high-level language, he *writes* a high-level language.

For those 6502 hackers, we offer the following preview to whet your ST appetite. And this is simply an hors d'oeuvre. Further details and definitions must be saved for later articles. If you can't wait, we recommend you pick up the previously mentioned **68000 Assembly Language Programming**, or **The 68000: Principles and Programming**, by Leo J. Scanlon, Howard W. Sams & Co., Inc.

continued on next page

68000 OVERVIEW

There are two operating modes in the 68000: User and Supervisor. Certain instructions in supervisor mode are not available in user mode. The supervisor mode is a protection against operator misuse, in sophisticated, multitasking systems. It should be interesting to see what Atari does with the supervisor mode.

Other niceties include built-in debugging aids, traps against illegal addressing and illegal instructions, a one-step trace mode, and seven levels of vectored interrupts. Most of these are only available from the supervisory mode.

DATA TYPES

Although the 68000 has a 16-bit data bus, meaning that 2 bytes of information can be accessed in one machine cycle, internally it can operate on five different types of data: bits, 4-bit binary coded decimal (BCD), 8-bit bytes (B), 16-bit words (W), and 32-bit long words (L). Because of this, byte data may be addressed at even or odd addresses, but words and long words must be addressed at even addresses. For example, three bytes in a row could fall at addresses $0004, $0005 and $0006, three words at $0004, $0006, $0008, and three long words at $0004, $0008, $000C.

The 68000 has 56 instructions and 14 addressing modes. This is very similar to the 6502. But there are 17 general-purpose 32-bit registers. Eight are considered data registers, seven are address registers, one is the stack pointer and the last is the program counter.

THE REGISTERS

All of the data registers are general purpose and can be used as index registers or counters. They can handle bytes, words, and long words. The address registers are primarily designed to hold addresses, but can be used as index registers. Unlike the data registers, they cannot handle 8-bit bytes.

The stack pointer can also be used as a general purpose address register. It is actually two registers and will contain different data depending upon whether you are in supervisor or user operating mode.

The last 32-bit register is the program counter and, although it is a 32-bit register, only 23 of the bits are used. Since instructions consist of words instead of bytes, the counter can access a range of 8M words, or 16,777,216 bytes. 6502 programmers will feel like a gnat in the Houston Astrodome.

STATUS REGISTER

The last register in the 68000 is the 16-bit status register, which is divided into two 8-bit bytes. The lower 8 bits are for the user mode and the upper 8 for the supervisor. Not all available bits are used. The user flag bits are:

BIT	SYMBOL	CONDITION
0	C	Carry
1	V	Overflow
2	Z	Zero
3	N	Negative
4	X	Extend
5–7	(Unused)	

Supervisor status flag bits 8 through 9 are used in various combinations to signal interrupt priority for the seven levels of interrupt. The 13th bit switches the modes between supervisor and user, and the 15th bit places the 68000 in trace mode. Bits 11,12 and 14 are unused.

ADDRESSING MODES

As stated above, there are fourteen addressing modes:

1. Data register direct
2. Address register direct
3. Register indirect
4. Register indirect with post-increment
5. Register indirect with pre-decrement
6. Register indirect with displacement
7. Register indirect with index
8. Absolute short
9. Absolute long
10. PC relative with displacement
11. PC relative with index
12. Immediate
13. Quick Immediate
14. Implied register

Given the number of registers and data types, the flexibility of register use, and the amount of indirection indicated in the address modes, there is incredible power available in the 68000.

MC68000 INSTRUCTIONS

Table 1 is a chart of the 68000 instruction set mnemonics with brief definitions.

Some instructions will be familiar to 6502 programmers, but many will be completely alien. There are no LDAs or STAs for example, because the 68000 is not accumulator bound. There is the remarkable MOVE which will move anything from anywhere to anywhere else.

Programming syntax for the 68000 on currently available assemblers is identical to popular 6502 assemblers, in that each line consists of:

Line number (Label) Mnemonic (Operand) (Comment)

The mnemonic field, however, may contain a three, four or five letter mnemonic, and instructions can occupy from one to five *words* in memory.

O BRAVE NEW WORLD

We hope this article has generated more questions than answers. Atari owners have been waiting a long time for The New Machine. It's here at last. I, for one, can't wait to get my hands on it.

68000 Assembly Language Programming
by Kane, Hawkins & Leventhal
Osborne/McGraw-Hill
2600 Tenth Street
Berkeley, CA 94710
(415) 548-2805
$18.95

The 68000: Principles and Programming
by Leo J. Scanlon
Howard W. Sams & Co.
4300 West 62nd Street
Indianapolis, IN 46268
(317) 298-5400
$15.95

Table 1
Instruction Mnemonics

Mnemonic	Description
ABCD	Add Decimal with Extend
ADD	Add
AND	Logical AND
ASL	Arithmetic Shift Left
ASR	Arithmetic Shift Right
B_{CC}	Branch Conditionally
BCHG	Bit Test and Change
BCLR	Bit Test and Clear
BRA	Branch Always
BSET	Bit Test and Set
BSR	Branch to Subroutine
BTST	Bit Test
CHK	Check Register Against Bounds
CLR	Clear Operand
CMP	Compare
DB_{CC}	Test Cond, Decrement and Branch
DIVS	Signed Divide
DIVU	Unsigned Divide
EOR	Exclusive OR
EXG	Exchange Registers
EXT	Sign Extend
JMP	Jump
JSR	Jump to Subroutine
LEA	Load Effective Address
LINK	Link Stack
LSL	Logical Shift Left
LSR	Logical Shift Right
MOVE	Move
MOVEM	Move Multiple Registers
MOVEP	Move Peripheral Data
MULS	Signed Multiply
MULU	Unsigned Multiply
NBCD	Negate Decimal with Extend
NEG	Negate
NOP	No Operation
NOT	One's Complement
OR	Logical OR
PEA	Push Effective Address
RESET	Reset External Devices
ROL	Rotate Left without Extend
ROR	Rotate Right without Extend
ROXL	Rotate Left with Extend
ROXR	Rotate Right with Extend
RTE	Return from Exception
RTR	Return and Restore
RTS	Return from Subroutine
SBCD	Subtract Decimal with Extend
S_{CC}	Set Conditional
STOP	Stop
SUB	Subtract
SWAP	Swap Data Register Halves
TAS	Test and Set Operand
TRAP	Trap
TRAPV	Trap on Overflow
TST	Test
UNLK	Unlink

MANIPULATING STRINGS

More power and speed from BASIC



by BRIAN WEISS

Learn how to manipulate the Variable Name Table and Variable Value Table in your BASIC programs. Professional programmers use these powerful techniques to add speed and variety to their code. The BASIC listings work with all Atari computers of any memory size.

If you program in BASIC, you've almost certainly used string variables to store and manipulate character strings. In this article, we'll see how Atari BASIC handles string variables. Then we'll show you how to use this information to harness the impressive power and speed of string manipulation in Atari BASIC.

These techniques come in handy whenever you have to move or modify a large area of memory. Some possible applications are: redefining character sets, changing display lists, manipulating machine language programs from BASIC, moving players in Player/Missile graphics, and changing screen memory for animation or page flipping.

VARIABLE TABLES

Atari BASIC uses two tables to keep track of the variables in your program: the Variable Name Table and the Variable Value Table.

The Variable Name Table holds the names of all the variables used in your program. It also tells the computer if a variable contains a string, a number, or a numeric array.

The Variable Value Table records the size and contents of each variable.

Whenever you use a new variable, whether it's in a program or a direct command, the computer updates both tables. Moreover, when you SAVE a program to disk or cassette, the tables are saved as well. Both tables will remain in memory until you LOAD another program, issue a NEW command, or turn off the computer.

FINDING TABLES

The location of these tables in memory depends upon the length of your program. To find them, we check the contents of four special

LINDA TAPSCOTT

memory locations which point to the start of the tables. These are:
VNTP=PEEK (130) + PEEK (131) * 256
VVTP=PEEK (134) + PEEK (135) * 256

VNTP is the starting address of the Variable Name Table and VVTP is the starting address of the Variable Value Table.

The Variable Name Table stores the names of all variables in the order they appear in your program listing. It also stores the type of each variable (string, numeric, or numeric array).

Listing 1 displays the contents of the Variable Name Table in a readable format. The subroutine in lines 1000 through 2005 can be appended to any BASIC program for a listing of the variables used. Type in Listing 1, check it with TYPO II and SAVE a copy.

INTERPRETING TABLES

The Variable Value Table tells the computer where to find the contents of each variable. It also contains the size of each variable. It uses eight bytes per variable to store this information. We'll refer to these bytes as byte one, byte two, etc.

Byte one determines whether the variable represents a string or a number. (A 129 in this location would denote a string variable.)

Byte two is a reference number (0 through 127) assigned to that variable. This is the number which the computer uses to identify each variable.

Byte three and byte four tell the computer where it can find the contents of the variable.

Byte five and byte six contain the length of the variable, and byte seven and byte eight contain the maximum size of the variable. In the case of string variables, this value is equal to its dimensioned length. For example, consider the BASIC line:

```
10 DIM A$(12):A$="ABC"
```

In this case, A$ is three characters long, but its maximum size is 12.

Interpreting bytes three through eight requires an understanding of how the computer stores numbers in memory locations. A single memory location can only store numbers between 0 and 255. The computer breaks up larger numbers into two parts and uses two memory locations to store them.

The first location contains the number of 256's in the number. Programmers refer to this as the "high byte" of a number. The second location, the "low byte," contains the rest of the number.

To calculate the size of a string variable, we multiply byte five by 256 and add byte six. We used the same method to calculate the starting addresses of the Variable Name Table (VNTP) and the Variable Value Table (VVTP) at the beginning of this article.

Finding the location of a variable in memory is slightly more complicated. Strings and arrays are stored in the String and Array Table. Byte three and byte four contain a pointer, or "offset value" used to locate a variable in this table.

First, multiply byte three by 256, and add byte four to determine this offset value. Now, add this to the starting address of the String and Array Table. This address can be found with the equation: STARP= PEEK (140) + PEEK (141) * 256. The final value is the actual address of the string. You could also find this address with BASIC's ADR function.

The location of the String and Array Table varies with the size of your BASIC program. For this reason, references to this table should only be done under program control, not through direct commands.

Listing 2 will display the values of byte one through byte eight for each string variable in the Variable Value Table.

Listing 3 takes the same information and interprets it for you, printing the actual size and location of each string variable in the program. Both of these listings should be checked with TYPO II and SAVEd before you RUN them.

LOCATING VARIABLES

Searching through the variable tables for information about a particular variable is a complicated, error-prone, and often unnecessary procedure. We can easily design and build our own variable tables, if we follow one simple rule: The order of variables in the tables must be the same as the physical order of the variables in your program. Consider the following program:

```
10 DIM A$(3):A$="ABC"
20 GOSUB 900
30 DIM B$(3):B$="DEF"
40 END
900 DIM C$(3):C$="GHI"
910 RETURN
```

In this program, A$ will be the first variable in the variable tables because it's the first variable in the program. B$ will be the second, and C$ will be the third. (Note that the computer builds the variable tables according to the physical order of variables in your program, NOT the logical order.)

If you want to manipulate a variable through the variable tables, it's wise to declare that variable in the first line of your program. This places it at the top of your variable tables.

Remember to type NEW before typing in your program to assure that no information is left in the tables from earlier programs or direct commands.

If you forgot to do this, you can LIST the programs to disk or cassette, type NEW, and ENTER the program again. Do not use SAVE and LOAD since these commands save the variable tables along with the program.

CHANGING VARIABLES

Both variable tables are in RAM which means that their contents can be changed using BASIC's POKE statement.

For example, suppose we wanted to change the name of the first variable used in Listing 1 from A$ to B$. Since A$ is the first variable used, it will be at the beginning of the table and its name will be in location VNTP. LOAD in Listing 1 and then type:

```
POKE VNTP, ASC ("B")
```

Now LIST the program. All references to A$ are now B$.

continued on next page

VALUE TABLE CHANGES

More powerful effects can be achieved by changing the Value Table. By changing a string variable's entry in this table, we can position it anywhere in memory. We can also change its size.

If we place our string variable in a region of memory used for other purposes, we can use it to change those memory locations. For example, if we relocate a string variable to an area reserved for Player/Missile graphics, we can control the players with several well-placed POKEs to the variable tables.

To do this, of course, you'd need a working knowledge of Player/Missile graphics, and that's beyond the scope of this article. Instead, we'll relocate a string variable to screen memory. In this way, we'll change what's on the screen by changing the appropriate entries in the Variable Value Table.

Recall that the Variable Value Table contains information about the location of string variables. This information is kept in bytes three and four in the table. In Listing 3, A$ is the first variable dimensioned. Since its byte one value is in memory location VVTP, its byte three value can be found by PEEKing (VVTP + 2), and its byte four value is in (VVTP + 3).

SLIDING STRINGS

Let's move A$ to screen memory. The address of the beginning of screen memory is calculated in line 100 of listing 3, and stored in the variable SCRN.

Next, we use SCRN to calculate A$'s new byte three and byte four values for the Variable Value table. Use the following formulas:

Offset= (new location)–
(PEEK(140)–PEEK (141) *256)

Byte four = INT (Offset/256)

Byte three = Offset–
Byte four * 256

Lines 150–170 perform these calculations. These new values are POKEd into the Variable Value table in line 200. These POKEs slide the contents of the string variable into screen memory.

Lines 250–290 change the variable's size by altering the values for bytes five and six in the Variable Value Table. We can calculate the new values with these formulas:

Byte six = INT (size/256)

Byte five = size–byte six * 256

In this example, we use a size of 400 bytes. This allows us to control the top ten lines of the Graphics 0 display screen.

MODIFYING MEMORY

Once relocated, a string variable can be used to modify the area of memory it occupies. Manipulating the string contents alters the contents of the memory locations. An example is the line:

A$="ABCD"

This will put a 65 in memory location SCRN, 66 in SCRN+1, 67 in SCRN+2, and so on. Since we are dealing directly with the screen, we must use internal character codes rather than ATASCII. In line 400 of Listing 3, A$ is completely filled with CHR$(0), the ATASCII "heart" character. This puts a 0 in the first 400 locations of the screen memory area, and 10 blank lines are on the screen.

Line 420 puts the word "HELLO" on the second line of the screen by placing the characters "(%,,/" into A$, and then sliding A$ to screen memory. In this example, when a 40 (ATASCII code for a left parenthesis) is POKEd into screen memory, the letter "H" appears on the screen.

The speed you can achieve with this "string sliding" is rivaled only by machine language. Imagine the graphics effects possible!

FOUR STEPS

Four basic steps are needed for positioning string variables in memory:

1. Dimension the variable in the beginning of your program.

2. Calculate VVTP, the starting address of the Variable Value Table, from locations 134 and 135.

3. Select a new memory location for your string variable, break the address into low and high bytes, and POKE these new values into the Variable Value table at byte three and byte four.

4. Calculate the new size of the string variable and place these values into byte five and byte six. Byte seven will contain the new value you need for byte five, and byte eight will contain the value you need for byte six.

Once the string variable is positioned, characters in the string will correspond to numbers in the memory locations. This method can be used to place numbers in memory by placing characters in the string. You can also to read numbers from memory by reading the string.

A string can even be positioned in the area of memory where a program is stored—resulting in a program that can write other programs! As you learn more advanced programming techniques, you'll discover many more uses for relocating string variables.

Brian Weiss is a computer science major at the University of Maryland and has been programming with the Atari for four years.

Listing on page 66.

lazy loader

Extra-convenient menu program

by FRANK WALTERS

This utility presents you with a menu of your BASIC programs. One-touch commands permit RUNning of SAVEd programs or ENTERing of LISTed programs. Works on all Atari computers with 24K memory, BASIC and a disk drive.

When you're wallowing in dozens of floppy disks, it's comforting to to immediately determine what programs lie on each disk and run them, without switching between BASIC and a DOS menu.

What you need is a menu program. 'What?", you say, "Another menu program?"

Ah—but Lazy Loader is different. It allows you to enter LISTed BASIC programs, reads up to four disk drives, and displays up to 51 files with horizontal scrolling.

USING THE PROGRAM

Type in Listing 1 and check it with TYPO II. SAVE it to a disk under the file name LAZYLOAD.BAS.

You can RUN the program right now and see all the BASIC files on your disk. Or, you can have the menu automatically boot with the disk by including an AUTORUN.SYS file. (See accompanying article for more information about AUTORUN.SYS files.

The file names will appear in a column in center screen. If you have more than 17 files, the column will scroll to the left one line at a time as a second column appears.

Because Lazy Loader is designed to handle BASIC programs, it will not display files with the extenders .SYS, .OBJ, .EXE, or .DAT.

Each file name is displayed with a letter, from capital A to Z, then lower case a to z.

To run a program, strike the corresponding letter. When choosing a file with a lower case letter, be sure that you are in lower case mode. This status will be displayed in the upper left corner.

If you want to see the file names of a different drive, choose the appropriate drive number (1–4).

ENTER OR RUN

In addition to RUNning tokenized or SAVEd programs, Lazy Loader ENTERs untokenized, LISTed files. Again, just press the key corresponding to the program you want. You don't need to press [RETURN].

If an error should occur, its number appears in the upper right corner. Press [RETURN] and the program will continue.

To read a new disk in the same drive, swap disks and press [RETURN]. To run DOS from drive 1, press [CTRL D].

Frank Walters, a retired Air Force fighter pilot, is a one-and-a-half year veteran of the Atari computer. He has also been a practicing amateur magician for 35 years.

Listing on page 68.

ALL ABOUT AUTORUN.SYS

There's not much point to using a menu program like Lazy Loader without an AUTORUN.SYS file—which will AUTOmatically RUN another program when your Atari is turned on. Here are the details of how it works and how to make one for Lazy Loader.

When you turn on your computer, it automatically goes through an initialization process called "coldstart". This is a byte-by-byte checking procedure that sets various statuses and values in the computer's central processor and other chips.

This overall process, called "booting" or "booting up" your computer, checks how much RAM is available, what peripherals are attached and turned on, and so on.

If a disk drive is on when you power up, the computer boots whatever disk is in the drive. If that disk contains the Atari Disk Operating System (DOS), it will be booted

continued on next page

into the Atari's memory from drive 1.

As part of its regular procedures, the DOS program looks for a binary file named AUTORUN.SYS. If AUTORUN.SYS is present, it is loaded into memory and RUN. Otherwise, DOS continues with other operations such as running the DUP.SYS file.

CREATING AUTORUN

The AUTORUN.SYS file must be a machine language program, but we can use a BASIC program which will create an AUTORUN.SYS file. In this issue's Listing Section and on the monthly Antic Disk, you'll find such a program, called ARSMAKER.BAS, (for AUTORUN.SYS MAKER). We found ARSMAKER.BAS in David Mentley's invaluable **ABCs of Atari Computers** (available from the Antic Catalog) and it is reprinted by permission of the publisher, Datamost.

Type in Listing 2, check it with TYPO II and SAVE a copy.

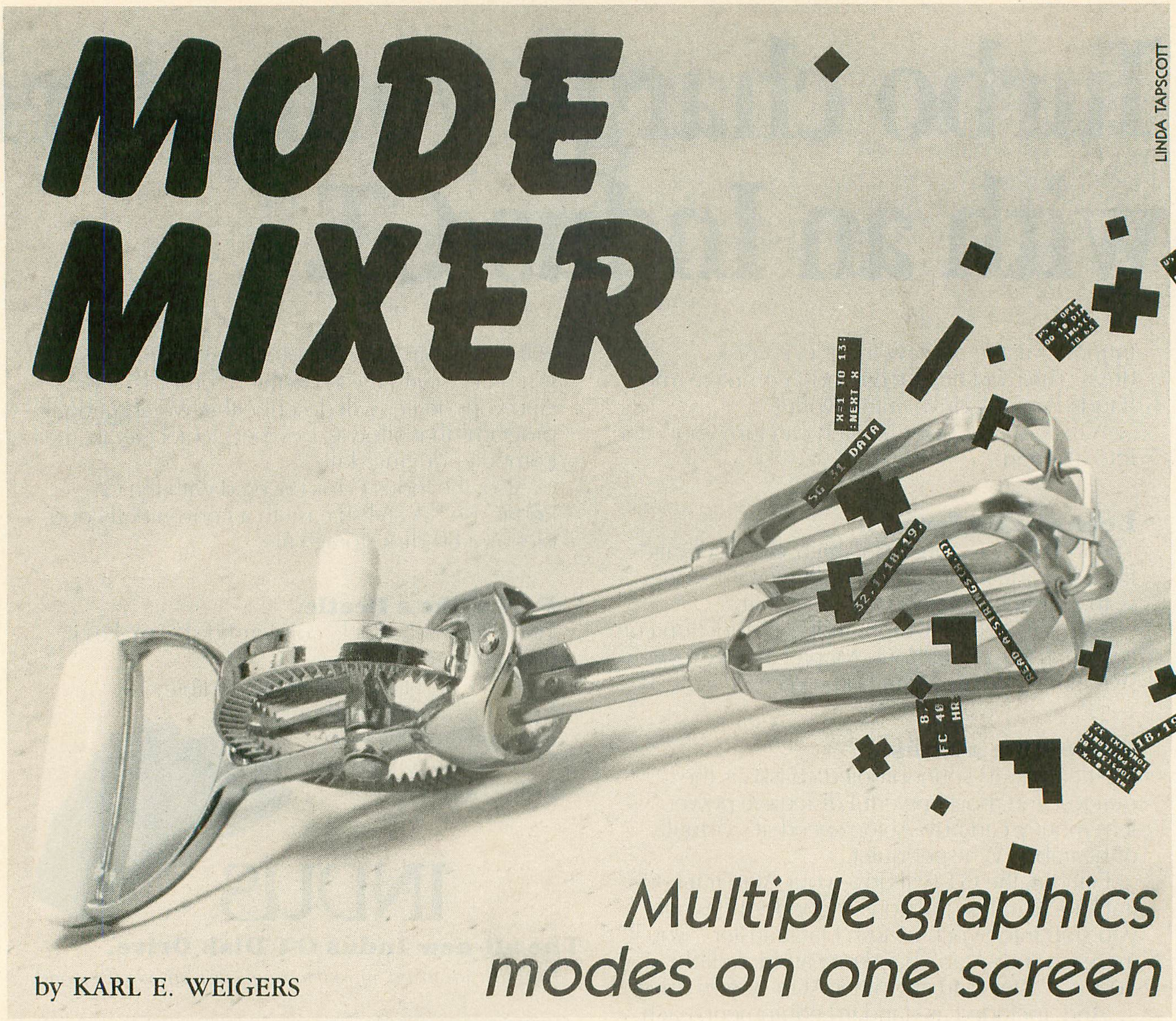
ARSMAKER.BAS creates an AUTORUN.SYS file that automatically runs any program of your choice. When you RUN ARSMAKER .BAS, the program will ask you to "Enter filename to auto run." At the ? prompt, type the file name, in this case the name you've given Lazy Loader, followed by a [RETURN]. (Don't type device code D:). You'll need to do this only the first time you RUN ARSMAKER.BAS on a given disk.

You now have a disk with DOS 2.0S, Lazy Loader, and a file named AUTORUN.SYS. When you boot this disk, the Atari automatically loads the DOS program, which automatically loads the AUTORUN.SYS file, which automatically runs Lazy Loader.

If you have an Antic Public Domain Disk or a monthly Antic Disk, here's a shortcut. Use DOS to change the name Lazy Loader or any other file to MENU. You can use the AUTORUN.SYS program that's included on each disk, since it will run any program named MENU. —M.C.

Listing on page 69.

MODE MIXER



Multiple graphics modes on one screen

by KARL E. WEIGERS

Mode Mixer is a utility program for easily creating custom display screens that use multiple graphics modes, and then saving them to disk for later use in your own programs. Written in BASIC, Mode Mixer requires a disk drive and will run on all Atari computers with 32K, depending on screen use.

DISPLAY LIST

The Atari's video display is created by a special microprocessor chip called ANTIC. (Yes, as in the title of your favorite magazine.) ANTIC can present information in 14 different graphics modes, six for text and eight for plotted graphics.

Only nine of these modes are available directly from BASIC on the old Atari 400 and 800 computers, but 13 of them can be accessed in BASIC with the XL models. The graphics modes differ in their horizontal and vertical resolution, number of colors shown, and amount of memory consumed (*Table 1*).

The program which tells ANTIC what kind of display to show is called a "display list". A nice introduction to display lists and mixed-mode principles is found in "Display Lists Simplified" (**Antic**, Feb/Mar 1983). "Unlocking the 56 Graphics Modes" (**Antic**, Sept. 1984) illustrates all these modes and provides more information about the Atari's video display.

A BASIC graphics call will only give you a screen of one mode (with the exception of windows). To construct a mixed-mode display, you must create a display list to tell ANTIC how many "mode lines" of each desired graphics mode to show. The resulting screen is made up of several horizontal bands or segments, one band for each block of lines in a particular graphics mode.

The display list also contains some other information, and a few tricks have to be played to print or plot correctly in the different segments. For each graphics mode, one mode line consists of a specific number of horizontal TV "scan lines", ranging from 1 for ANTIC mode 15 (GRAPHICS 8) to 16 for ANTIC mode 7 (GRAPHICS 2). A complete screen must contain 192 scan lines. Keep these numbers in mind as you use Mode Mixer.

USING MODE MIXER

Type in Listing 1, check it with TYPO II, and SAVE a couple of backup copies. RUNning the program shows you a menu. Just press a number key to choose a menu selection; it is not necessary to press [RETURN]:

1. **Create a New Display:** Begin with this option. You will see a screen which prompts you to enter the ANTIC graphics mode (2–15) for the first segment. Then you can specify the number of mode lines for that segment. Continue this process for all the segments in your display. The table at the top of the screen keeps a running description of your display as you compose it. Keep an eye on the number of scan lines remaining, which goes down each time you describe a new segment. The goal is for this number to be zero when your display is complete.

You can enter up to 16 segments per display, combining the available graphics modes any way you like. When prompted for a graphics mode, enter the ANTIC mode number from *Table 1*, not the usual BASIC graphics mode number. To cancel an entry, enter zero for the number of mode lines. When you have completed the description of your display, and wish to return to the menu, enter zero for the next mode number.

Notice that selecting option 1 at the menu erases forever any display already in memory.

2. **Change the Current Display:** Use this option to modify the display currently in memory, whether you just created it or loaded it from the disk. You will be prompted for the segment number to modify, and then for the new graphics mode and number of lines.

If the number of scan lines remaining is greater than zero and the number of segments in the display is less than 16, then you can add new segments to the bottom of the display. When prompted for the segment number, just enter the number one greater than the last segment number already in the display. Also, you can erase a segment by setting the number of mode lines for that segment to zero.

When you have finished editing the display, enter zero for the segment number to change. You will return to the menu.

3. **See the Current Display:** This function actually generates the display list from your description of the mixed-mode screen. The TV screen will be black briefly while the calculations are done, and you will hear a tune reminiscent of Woody Woodpecker when the computer is done. At that point, you will see the mixed-mode display. Admittedly, it isn't very interesting: just bands of blue (for ANTIC modes 2, 3, and 15) and black (for all other modes). In fact, you may not see anything at all. Rest assured that all is well, however. After you hear the tune, press any key to return to the menu.

4. **Save Current Display on Disk:** This part of Mode Mixer writes a program for you and stores it on the disk. First you will be asked to supply a file name. Use any legal Atari disk file name, without extender. Unless otherwise specified, Mode Mixer assumes you mean disk drive one, and it assigns an extension of ".DSP" to the file name you give. If you try to save the display before you viewed it with menu option 3, you will go through step 3 anyway.

The resultant file will be a BASIC program, stored in LISTed form. We will talk later about how to use this program.

5. **Load a Display from Disk:** Here you can retrieve a display from the disk for further editing. MODE MIXER shows you a list of all the files with extension ".DSP" and you type the name of the file you wish to load. Enter a zero if you decide not to load anything after all.

It takes several seconds for the program to read the file and reconstruct the display description. When this process is complete, you will be at the "Change the Current Display" screen. Notice that using option 5 replaces any existing display in memory with the one you load from the disk.

6. **Done With This Program:** Choose option 6 to leave Mode Mixer and return to BASIC.

STORED DISPLAYS

To use a stored display, type NEW to clear out any program already in the computer's memory, then type ENTER "D:filename.DSP". You can now RUN this program to recreate the display as you described it to MODE MIXER. This program consists of several parts:

1. Line 10, which calls a subroutine at line 30000.

2. Several statements which set up the various segments in your display. Each begins with a POKE 87,n command, where n is a BASIC graphics mode number.

3. An END statement at line 29999.

4. A block of statements beginning at line 30000 which set up the display list needed by your mixed-mode screen.

5. A subroutine beginning at line 31000 which controls memory allocation for each display segment.

The point of all this work so far is to help you get some fancy displays on the screen. You accomplish this by thinking of each segment as a separate little screen. Following each of the POKE 87,n statements you can insert any appropriate text or graphics display commands. For text modes (ANTIC 2–5) use POSITION and PRINT #6; statements.

In the graphics segments (ANTIC 6–15), use PLOT and DRAWTO commands. The upper left corner of each screen segment is location 0,0. Be sure not to plot or print outside the allowable horizontal and vertical boundaries

continued on next page

of a segment, based on the number of mode lines you have in each segment.

You can easily combine a program written by Mode Mixer with any other BASIC program. Use the ENTER command to retrieve the display program file from disk and merge it with another program already in memory. Remember that line numbers from an ENTERed file replace any statements with the same line numbers in the existing program when you do a merge operation.

AN EXAMPLE

Listing 2 is a sample program written by Mode Mixer. The display specified has six segments: 4 lines of ANTIC mode 6; 4 lines of ANTIC 2; 20 lines of ANTIC 14; 4 lines of ANTIC 5; 20 lines of ANTIC 15; and 6 lines of ANTIC 10.

Type in Listing 2 and LIST it to disk. Next, type in listing 3 and SAVE it. With listing 3 in memory, ENTER Listing 2 to merge the two programs, then RUN it to see a nice demonstration of the kind of complex displays you can produce using Mode Mixer and a bit of your own creativity. NOTE: Press [SYSTEM RESET] after running any program created using Mode Mixer.

OTHER TIPS

You can change the color registers as usual in a program with a mixed-mode display. (**Atari Color Graphics** or **Atari Graphics and Arcade Game Design**, available from the Antic Catalog, fully describe what the different color registers do in each BASIC graphics mode.) Using display list interrupts (see "More Interrupting" in **Antic**, Dec. 1983) to get different colors in different segments will help create truly dazzling displays.

To get a uniform background color for displays in which ANTIC modes 2, 3, or 15 are mixed with the other modes, use a SETCOLOR 2,H,L statement, where H and L are the hue and luminance of the desired background color (0,0 is the default). Unfortunately, this also makes any graphics which use a COLOR 3 statement invisible.

I haven't forgotten about the three GTIA graphics modes, BASIC modes 9, 10, and 11. These all use the same display list as ANTIC 15, so just create a segment with mode 15 with Mode Mixer. Then change the POKE 87,8 statement for that segment to a POKE 87,9 (or 10 or 11, depending on the GTIA mode you want). You will need to use display list interrupts to alter the GTIA location at $D01B for only the appropriate segments.

The XL computers have a BASIC mode number for all ANTIC modes except 3. Programs written with Mode Mixer will work fine on the XL computers. However, you may wish to change the "n" in the POKE 87,n statements for segments of ANTIC 4, 5, 12, and 14 to the appropriate BASIC mode from *Table 1.*

Line 15 of Listing 3 (POKE 752,1) prevents stray cursors from appearing when printing in text segments of a mixed mode display.

Table 1

Atari Graphics Modes

ANTIC	BASIC	Scan Lines/ Mode Line	Mode Lines/ Screen
2	0	8	24
3	NONE	10	about 19
4	12 (XL)	8	24
5	13 (XL)	16	12
6	1	8	24
7	2	16	12
8	3	8	24
9	4	4	48
10	5	4	48
11	6	2	96
12	14 (XL)	1	192
13	7	2	96
14	15 (XL)	1	192
15	8	1	192

Table 2

Variables Used in Mode Mixer

INC	— increment counter for number of bytes per segment
MODE	— ANTIC mode for each segment
LINES	— number of mode lines for each segment
SLPER	— number of scan lines per mode line in each ANTIC mode
BPER	— number of bytes per mode line in each ANTIC mode
BAS	— BASIC graphics mode number to use for each ANTIC mode
B$	— general use string variable
A$	— general use string variable
FNAME$	— filename for loading or storing display
BL$	— string of blank characters
OFF	— value to disable a TRAP statement
NO	— line number of subroutine to make error sound
MEMORY	— line number of a subroutine to set screen memory locations for a segment
LOCS	— line number of a subroutine to calculate location of screen memory and display list
SPACE	— line number of a subroutine to wait for space bar to be pressed
DL	— starting location of display list
MEM	— starting location for screen memory
LMEM	— low byte of starting location of screen memory
HMEM	— high byte of starting location of screen memory

BYTE	— number of bytes of screen memory in a segment
CH	— general input variable
NSEG	— number of segments in a display
GRA	— ANTIC mode of current segment
NUM	— number of mode lines in a current segment
LEFT	— number of scan lines left in this display
A	— general input variable
CT	— counter for number of bytes in display list
OSMODE	— mode number of a segment as read from disk
MAX	— BASIC graphics mode number of segment in the display with the greatest screen memory requirement
FOURK	— next 4K boundary after beginning of screen memory
Y, I, J	— variables for loops, offsets, temporary calculations
LN	— line number of statement in file being saved to disk

Table 3

Mode Mixer Program Take-Apart

Line Numbers	Function
1–99	initialize variables and arrays
100–230	menu
500–520	subroutine to make error sound
550–560	subroutine to set new screen memory locations
600–610	subroutine to calculate screen memory locations
625–635	subroutine to pause until space bar is pressed
900–970	subroutine to set up display create/change screen
999–1070	create a new display; define mode for a segment
1080–1100	define number of mode lines for a segment
1110–1140	calculate number of scan lines left, update table
1999–2005	change current display routine starts here
2010–2040	show current display description
2050–2165	input segment number to change and make changes
2170–2190	update table describing display
2999–3080	load display from disk; show displays on disk
3090–3170	get file to load, open file
3180–3290	figure out no. of segments & display list bytes
3300–3400	figure out no. of mode lines per segment
5999–6050	see the current display routine starts here
6100–6140	figure out maximum memory requirements mode
6145–6290	create display list in page 6
6320–6360	show display, play tune, wait for keypress
6370–6380	branch based on whether user wants to store display
6999–7000	save display on disk routine starts here
7010–7080	get filename and open file
7100–7420	write statements to file which will recreate display
7500–7520	error handling for bad file open

Karl Wiegers is a frequent contributor to the pages of ***Antic****. His most recent work was "Touch Tablet Cursor" seen in our January 1985 issue*

Listing on page 63.

COMPUTER PALACE WE KNOW ATARI!

CALL US AT (503)683-5361 FOR INFORMATION

INDUS GT DISK DRIVE

- Free Software
- Double Density
- 1-Yr. Warranty



Our Most Popular Drive!

We recommend and sell more of these drives than any other. It's so quiet that you'll probably forget you are using a disk drive. Flip-up dust cover, LED readout, and FREE SOFTWARE (DOS XL, Word processor, Database, Spreadsheet, Programming book by *Datamost*)...This package is hard to beat!

New Low Price **$259.95**

TRIVIA QUEST

48K Disk
$39.95

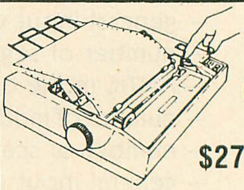
- Program Covers 4 Disk Sides



- Outsmart Your Friends
- Outwit The Dragon
- Join The Quest

A new concept in computer gaming. Intellectual challenge, strategy and arcade action. Each player assumes the role of a lord with a questing party of three characters. Complete the quest, earn the most gold by answering questions and battling the dragon. Win the favor of the king and thus, the game.

- **Utility Disk:** 1000 Additional questions plus create your own... **$24.95**



Super-5
CPA80
Printer

$279.00
+ FRT

A "best buy"! Features "square dot" print head with the best print quality for the money. 100CPS, bi-directional, graphics, and EPSON compatible. Plain or Pin-fed paper.

GAMES

Star Raiders	13.95
Joust	35.70
Robotron	35.70
Donkey Kong	27.90
Dig Dug	27.90
Defender	17.95
Ulysses	19.95
Tennis	13.95
Ultima II	52.50
Ultima III	39.95
Bounty Bob	44.90
Strip Poker	31.50
Bridge 4.0	22.50
Spy vs Spy	26.90
Castle Wolfenst	26.90
Poker Sam	22.50
Bristles	26.90
Cutthroats	31.50
Enchanter	31.50
ML Baseball	35.10
Chess 7.0	59.50
Checkers	44.90
Pitstop II	35.10
Computer Ambush	52.50
Broadsides	35.10
Kampfgruppe	52.50
Adventure Writer	26.90
Ali Baba	26.90
Return Hercules	28.90
Savage Island	17.90
Voodoo Castle	17.90
Secret Mission	17.90
Dimension-X	31.50
Zombies	31.50
Bruce Lee	35.10
Conan	35.10
Ghost Busters	26.90
Archon II	35.10

EDUCATIONAL

Hundreds of Titles Please call us.

BOOKS

Misc:	
6502 Programming	18.95
6502 Subroutines	17.95
Advd Pro Tech At	14.50
Advent w/Atari	14.95
Assem Lan Progr	15.95
Basic Atari Bas	15.95
Basic Atari Kids	12.95
Best Antic w/Dsk	24.95
Elementary Atari	14.95
Atari:	
Assembler	15.95
Basic Fast & Bet	19.95
Basic Source Bk	12.95
For Kids 8-80	15.95
Games & Rec	14.95
Graphics & Arc	16.95
Playground	9.95
Roots	14.95
Sound & Graphics	10.95
Users Encyloped	19.95
W/55 Programs	14.50
Compute:	
2nd BK Mach Lan	14.95
FBO Atari Graph	12.95
FBO Atari Games	12.95
FBO Atari	12.95
Great ADV Games	12.95
2nd Bk of Atari	12.95
3rd Bk of Atari	12.95

Super Mailer PLUS

Much More Than A Mailing List!

One of the most versatile data-base programs available.

Includes:

- FAST SEARCH (1 Sec. to find a name out of 1000)
- SEARCH ON ANY FIELD
- STATE CODE ABBREVIATION TABLE built-in
- SPECIAL CODE FIELD allows you to retrieve data from up to 12 codes
- PRINT LABELS 1, 2, or 3 up
- FAST SORT of your records by name in any of 8 fields (Sort 200 records in less than 40 seconds-true 3-D sort)
- MERGE files to create combination files
- DELETE DUPLICATES automatic or manual
- REDEFINABLE FIELDS

- **Maintain your book library...**
- **Organize your record collection...**
- **Index your recipies...**
- **Categorize your stamp collection...**
- **Unlimited applications!**

Features:

- LIGHTNING FAST RETRIEVAL
- FAST SORTS ON ANY FIELD
- SUPPORTS UP TO 4 DRIVES
- SINGLE OR DOUBLE DENSITY
- STORE ABOUT 1200 RECORDS PER DISK SIDE IN DOUBLE DENSITY
- MUCH MORE!

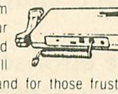
48K Disk $49.95

New Enhanced Version **2.0**

New! Mail Merge Utility for Atariwriter, Letter Wizard . . . **$19.95**

Now you can use your Super Mailer + records with *Atarwriter* and *Letter Wizard*. Use names and addresses to create form letters. Input special characters into the word processing programs to tell super mailer + where to put the information. It's as easy as 1...2...3!

NEW! COMPUTER PALACE B.B.S.



We are now running a bulletin board system as a service to our modem customers. You will be able to place an order with us by using your modem: just follow the screen instructions. We will be featuring download files such as public domain games, utilities, and program demos. We will have a section about the latest Atari products, news, and rumors... and for those frustrated adventurers, there will even be a section of hints and answers for the whole Atari adventure realm!

Give us a call! **1-503-683-7454** (6:00p.m. to 8:00a.m. PST)

BUSINESS

Assm Editor	31.50
Monkey Wrch II	29.95
Atariwriter	39.95
Letterperfect	87.90
Spell Wizard	44.90
ABC Compiler	69.95
B/Graph	62.50
Peachtree G/L	95.00
Peachtree A/P	95.00
Peachtree A/R	95.00
Home Accountant	67.50
Synfile	49.95
Syncalc	49.95
Syntrend	49.95
Synstock	35.10
Home-Calc	35.10
Mac/65	87.90
Action	87.90
Basic XL	87.90
Writer's Tool	87.90
Tele-Talk	44.90
Financial Cookbk	44.90
Basic Cartridge	29.90
Graphic's Gener	22.50
Microsft Basic 2	55.00

PARTS & MISC

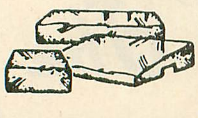
Disk Cleaner	13.95
800 Power Supply	24.95
10 Disk Mailers	6.50
10 Generic Disks	13.95
Ten Key Pad	39.95
Touch Tablet	54.95
Compu-Serve	39.95
Indus Dr Stacker	19.95
Oak Monitor Std	29.95
Printer Stand	24.95
Disk Bank(50)	15.95
Disk Bank(35)	9.95
Regular Joystick	9.30
Deluxe Joystick	19.95
Swivel Base	39.95
Joystick Cord	2.95
Joystick Handle	1.49
Joystick Board	2.49
J stick 12' Ext	9.95
Surge Protector	39.95
Printer Intrface	69.95
64K for 600XL	79.95
Relax	99.95
Sherlock	26.90

SPECIALS

Rally Speedway	19.95C
Shamus	9.95D
Pacific Cst Hwy	9.95D
Crossfire(T - D)	9.95
Match Racer	9.95D
Slime	9.95D
Bandits	9.95D
Jawbreaker	9.95D
Eastern Front	9.95T
Canyon Clmb(T - D)	9.95
Pool 400	14.95C
Speedway Blast	14.95C
Repton	9.95D
Wayout	9.95D
Threshold	9.95D
Stellar Shuttle	9.95T
Wiz of Wor	17.95C
Deluxe Invaders	17.95C
Gorf	17.95C
Miner 2049er	17.95C
Choplifter	19.95D
Lords of Karma	9.95D

D Disk T Tape
C Cartridge

Protect Your Equipment DUST COVERS



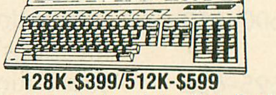
Custom fitted, attractive leather brown color:
- ATARI 400, 800, 600/800/1200XL, New XE&ST, 410, 810, 1050, 1025, 1027, CX85
- EPSON, GEMINI, PROWRITER printers
- INDUS, RANA, PERCOM, TRAK disk drives.

ONLY **$8.95** EACH | Additional covers ordered at same time ONLY **$7.95** EACH

Peachtree Software

Now, one of the most popular accounting systems is availale for Atari. *Back to Basics Accounting System* is a double entry, accrual accounting system consisting of three interactive packages for the small business: **General Ledger, Accounts Receivable and Accounts Payable.** A powerful system, it includes automatic posting, system generated mailing labels and password security. For the non-accountant, it comes with one of the most comprehensive manauals we have seen. For the expert, it will finally put your Atari in business. Requires 2 drives. **48K Disk**-System Package...**$195.00.** Each...**$95.00**

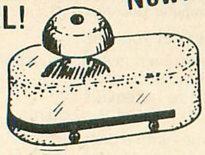
New ATARI 130/520ST Computers



128K-$399/512K-$599

The current flagship of the Atari Family has arrived, utilizing the speed of the Motorola 68000 CPU. With 128K or 512K, you will have power at prices you won't believe. And with a mouse, pull-down menus, windows, icon graphics and cut and paste features that allow you to integrate spreadsheet, text and graphic files; creative solutions have never been easier. The quantities are limited, so place your order now to get yours as soon as possible.

DISK NOTCHER SPECIAL! New!



Only $9.95

Now use both sides of your diskettes

Simply place the disk against the built-in stops and squeeze

FREE CATALOG

with any order ... or send $1 (refundable with first purchase)

You will receive the most comprehensive reference catalog available. Containing hundreds of software and hardware listings with illustrations and descriptions, our main catalog will give you the answers you need. Join our mailing list and receive free flyers with updates and special offers.



New Price...$29.95 **Cartridge**

Open up a whole new world of **BASIC** computing ease. 33 direct mode commands and functions. Also includes a machine language monitor to interact with the 6502.

Monkey Wrench II — Eastern House



Now Available for the XL!

COMPUTER PALACE

OPEN M-F, 9-6 Sat. 10-4 (Pacific Time)
2160 W. 11th Avenue Eugene, Oregon 97402

USE YOUR CREDIT CARD & CALL
Toll Free 1-800-452-8013
★ ORDERS ONLY, PLEASE ★
There's never a penalty for using your credit card!
For Information, Call (503) 683-5361

SHIPPING INFO: Minimum $2.90 Ground, $4.75 Air. Actual Cost depends on weight. Call (503) 683-5361 for information.
WARRANTY INFO: Everything that we sell is warrantied by the manufacturer. If any item purchased from us fails to perform properly when you receive it, call us at (503) 683-5361 so that we can assist you. No returned merchandise accepted without authorization. Defective software will be replaced with another copy of the same program, otherwise, no software is returnable.

beer party atari



by DR. JOHN C. FERGUSON

*"The Atari is just a game machine." Do those words set your blood aboil? Well, the staff at **Antic** is tired of them too, and spends a good portion of its time diligently combing the incoming submissions for practical applications programs. We receive a lot of disk directory programs, recipe file storers, mini word-processors, and other rehashed versions of old ideas. But now we'd like to award Dr. John C. Ferguson the Honorary **Antic** Unprecedented Application Program of the Year Award for his fine Beer Party Atari. A program whose time has come.* —ANTIC ED

This useful applications program will keep track of the best brew in a beer tasting party. The programming makes good use of custom display lists and character sets. So non beer-drinkers might wish to type it in and change the name to Rootbeer Party Atari. It is written in BASIC and will run on all Atari computers with 16K. A disk drive is recommended, but the article explains how to run the program without one.

My wife and I decided to have a party one evening for a fairly large number of workplace acquaintances. The trouble with such social gatherings is that "shop talk" tends to predominate and guests don't really get to relax and have fun. Our answer to this concern was to organize the party around the distraction of a "beer sampling" — to determine which brand of beer is best. The Atari provided an ideal tool to focus the group's attention towards finding an unbiased corporate answer to this important question.

We set up a table with five pitchers containing different beers. These were kept filled, out of sight, from cans of popular brands kept on ice. While the guests knew which brands of beer were involved, the pitchers were only labeled as "A", "B", "C", "D", or "E". The guests thus had to taste from each pitcher and give the mystery beer a rating. They could jot notes on a piece of paper to help themselves remember.

After sampling and rating each of the five beers, they then typed their evaluations into the Atari. The program I developed for this purpose made it easy for even the most computer-phobic in the crowd.

SOBERING INFLUENCE

The program provided a prompt for entering the rating of each beer, and then a chance to verify that all five were keyed in correctly. It then calculated the average accumulated score for each beer, saved the data to disk, and quickly showed a graphic display of how the different beer brands stacked up in the opinion of the judges.

A lot of guests were very surprised to see how swiftly the scores changed as more and more people entered their choices. It became almost like a horse race! If things got close, the numerical values of the average scores could be displayed by pressing [S] while the graph was onscreen.

VALUABLE DATA

Saving the data to disk after each set of entries was a feature added to the program to make sure that an accident didn't happen to spoil the accumulated results. It was fortunate that this precaution was included, because a power glitch did occur in the middle of my party and wipe out the program. However, I was able to quickly reload it, recall the accumulated data,

continued on next page

and continue on as if nothing had happened.

All in all, my beer party was a tremendous success. Everybody had a good time, and work worries were kept well out of mind.

THE PROGRAM

Type in the program, check it with TYPO II and SAVE a copy. Much of the program is internally documented with REM statements. When you RUN it, the computer will first ask you if you want to add to a previous file—that is, do you want to start with data saved from a previous run of the program. The first time your answer should be [N]. Note that if this choice is taken, any previous file of BEERDATA will be deleted and replaced with a brand new one. If you answer [Y] you are given a chance to insert the particular BEERDATA file disk you want to add onto before the program continues.

WITHOUT DRIVES

If you do not have a disk drive, you can still use the program without this feature. You should type REM after line numbers 110–160 and 1450. This will update your scores in memory without SAVEing them. It also preserves the code for possible future use when you do have access to a drive.

MODIFIED CHARACTERS AND DISPLAY LIST

Several programing "tricks" were used to produce the varied and interesting screen displays. First, some characters of the normal Atari font were modified to produce graphic representations of a beer stein and pilsner glasses. (See lines 10000–10200.)

Next, a customized Graphics 0 screen was produced by modifying the display list to show several lines of Graphics 1 and Graphics 2 (program lines 1030–1055). Now, when the altered characters, the normal text, and the special Atari control characters are all put together, in any of the the three sizes provided by the modified screen, the result is a really sophisticated display. This was produced with remarkably little code—Ah, the beauty of Atari!

NAME YOUR BRAND

When you use the program, you will want to select your own five brands of beer to compare. These should be reprogramed into lines 1550–1630. Note which beer corresponds to each code letter—you don't want to have these mismatched. For my first party I invested in a case of pilsner glasses and a sufficient quantity of five middle-line popular beers. Next year I think I will do it again with an international flavor—rating beers from five different countries.

Dr. John C. Ferguson is a Professor of Biology in the Department of Natural Sciences at Eckerd College in St. Petersburg, Florida. Dr. Ferguson has taught at Eckerd since 1963 and specializes in Marine Biology and Oceanographic Sciences.

Listing on page 62

SON OF INFOBITS

Now you can save and edit your Infobits files

by ANDY BARTON

This enhancement of the remarkably simple and popular file program from the December, 1984 ***Antic*** *provides an easy way to save, retrieve, edit and delete short notes and data. Written in BASIC, the all-new version is complete in this issue and will run on any Atari computer with a disk drive.*

In the December, 1984 ***Antic*** *we ran a program called "Info Bits" by Andy Barton, author of TYPO II. Published primarily to demonstrate how much can be accomplished with a small amount of programming, we described Info Bits as the "Simplest database ever!". We received many letters from readers who loved the program but wanted it just a little less simple.*

Specifically, they wanted to know how to edit or erase some of the many entries they were putting into their "Info Bits" data files. We contacted Andy and received the following update. —ANTIC ED

INFOBITS MODIFIED

Info Bits was originally written as a hasty last effort, after two rejections, to get something published. It was purposely kept short, simple and direct, with no frills added. In this spirit, a "Delete Entry" routine was not included. (And, to be honest, I didn't think about it until some time later when I found I had no way to correct a typographical error in one of my entries.)

TYPING THE PROGRAMS

Listing 1 is the new Info Bits. (Filename: INFONEW.BAS.) Those of you who typed in the original program may notice a few matching lines, but there are many changes so you will probably be best served by typing in all of the new listing. Remember to check your typing with TYPO II and SAVE a copy of the program before running it.

continued on next page

Listing 2 is a brief program which corrects a bug in the original Info Bits program by modifying its INFOBITS. FIL data file. If you have no Info Bit files yet, you can ignore this listing. Those who wish to use their previous INFOBITS.FIL entries with the new Info Bits must use this program to reconfigure their old files.

After typing in Listing 2 with a TYPO II checkup and SAVEing a copy, place the disk with a *copy* of your INFOBITS.FIL on it. RUN listing 2 and it will remove the leading 10 blank spaces in front of each entry on the file. Note: Listing 2 was kept purposely short and contains no error trapping. Make sure your disk contains the INFOBITS.FIL before you run the program.

USING THE PROGRAM

Info Bits is a mini data base. You can type in notes and references and then retrieve them by searching for a key word or phrase—which can be anything in your entries.

Before you can use Info Bits, you must create an empty file named INFOBITS.FIL on whichever disk is in drive one. To create the file, type in the following:

```
OPEN #1,8,0,"D:INFOBITS.FIL":
CLOSE #1 [RETURN]
```

After you've done this, RUN the new INFOBITS and you'll see a menu with three choices: 1.) ADD TO FILE, 2.) SEARCH FOR ENTRY, and 3.) DELETE/(EDIT) ENTRY.

ADD TO FILE

Press [1] [RETURN] and type in a couple of entries, pressing [RETURN] to mark the end of each entry. You don't need any special characters to mark off fields, simply type in a space or two. Each entry can be as long as 119 characters.

When finished making entries, press [RETURN] at the "TYPE ENTRY:" prompt and you will return to the menu.

SEARCHING FILES

To search for data, type [2] [RETURN] from the menu and you will see SEARCH FOR:. You may type in a word or portion of a word or a complete sentence. Enter your search criteria in capital letters and Info Bits will ignore case distinction in the data it searches.

After entering your search criteria, press [RETURN] and Info Bits will display on screen every entry that contains an exact match of your criteria. If there is more than one screen, the program will pause and prompt you to press any key for more. If you wish to see all your entries, enter ALL at the prompt.

DELETING AND EDITING

Type [3] [RETURN] to select DELETE/(EDIT) ENTRY. The screen is cleared and you are given 3 choices. First type [S] [RETURN] to search for the entry to be deleted. Repeat this step as often as necessary until the entry you wish to delete is the last one displayed.

Type [D] [RETURN] to delete the last entry displayed. The actual entry being deleted will be displayed on the screen. If you are satisfied with the deletion simply press [RETURN] to return to the main menu. If not, press [Y] [RETURN] to edit it. And when you are ready to re-enter it again, press [RETURN].

You are now in the ADD TO FILE routine of the main menu where you may continue adding entries when asked to TYPE ENTRY:. Or simply press [RETURN] on the blank line (an absolutely necessary step to properly close the disk file) and return to the main menu.

The only safe place to exit Info Bits (turn off the computer or load another program) is from the main menu. If you exit in the ADD TO FILE or DELETE/EDIT routines the disk file will not be properly closed and part or all of it could be lost.

UPPER AND LOWER

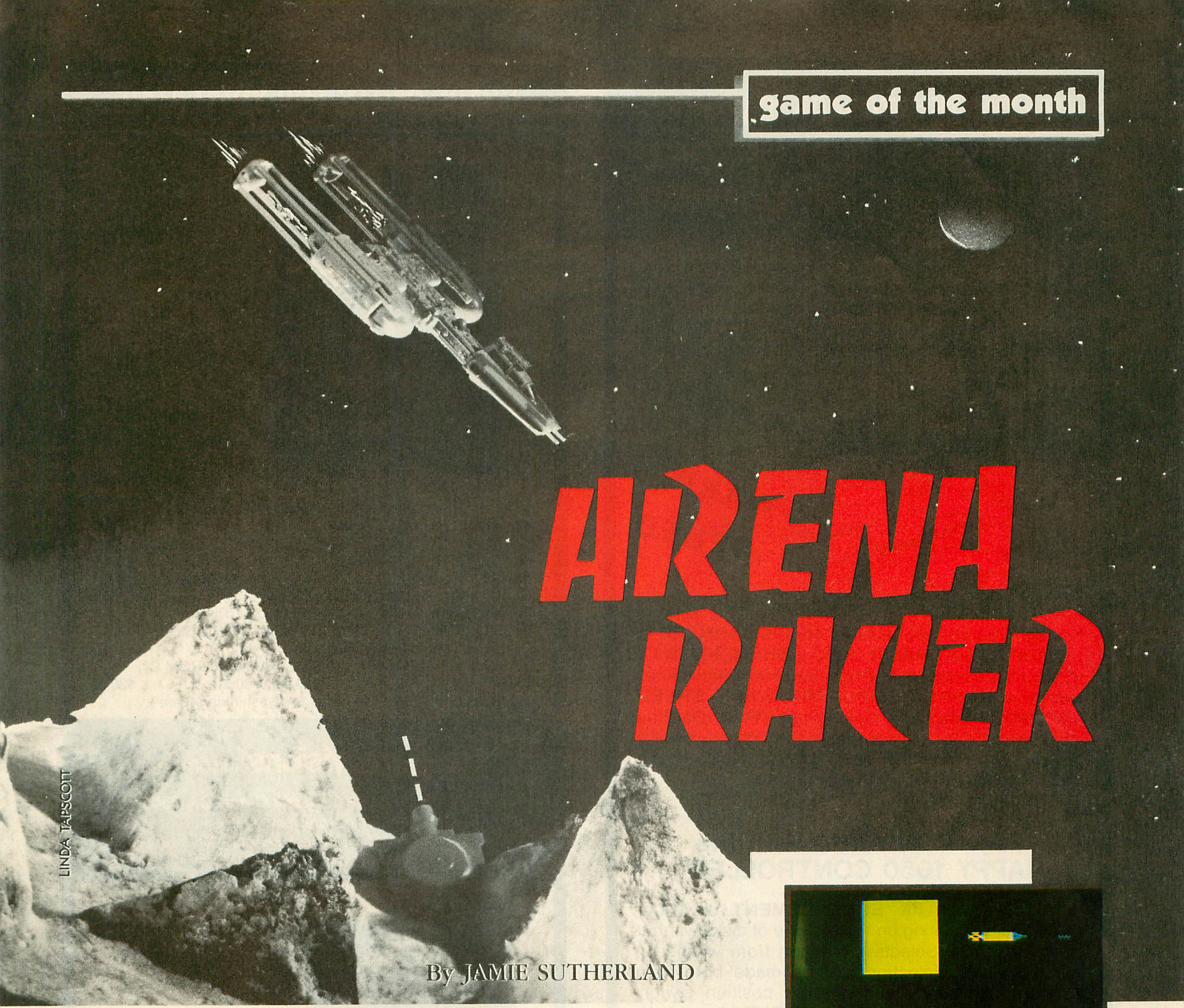
Included in this version of Info Bits is a short addition to the machine language routine that allows it to ignore the differences between lower and upper case letters. This allows you to type entries in caps and lower case letters for aesthetic or other reasons and later find them without having to remember which way you entered them. However, your search strings must be in capital letters.

If you prefer the original version which differentiates upper and lower case letters, simply replace the 5th through 10th numbers in DATA line 2002 with 234,234,234,234,234, 234.

Given the infallibility of computers (ha,ha), not to mention that of the operator or programmer, I highly recommend that you regularly make a backup copy of your disk file (INFOBITS.FIL).

*Many **Antic** readers owe Andy Barton a vote of thanks for his TYPO II line-by-line proofreading program, which we've been using with all our BASIC listings since January.*

Listing on page 67.



LINDA TAPSCOTT

ARENA RACER

By JAMIE SUTHERLAND

Arena Racer is 15 scrolling levels of fiercely challenging maze action. Can YOU fly a speeding Huntercraft through labyrinthian caverns and avoid getting blasted apart by all those Laser Cannons? This unusually fast BASIC program works on all Atari computers of any memory size.

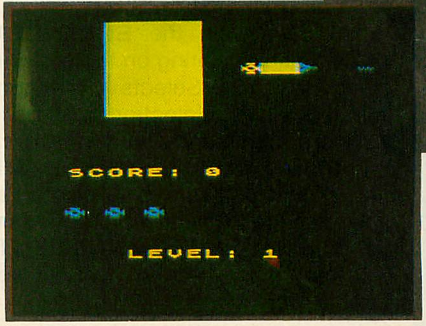
Uh-oh! There's an intergalactic war on and you got shot down onto a rather nasty planet. The local enemy warlord gets plenty of entertainment from Elite Fighter Pilots like you—when they're unlucky enough to fall into his hands. (Yes, you are a rocket ace. You only lost that last dogfight because you were so vastly outnumbered.)

You are tossed into an unarmed Huntercraft and sent down to a horror known as. . .the Arena! This Arena is a vast maze of caverns. On each of the caverns' 15 levels, you must collect four glowing orbs, which have been hidden and protected. But. . .as you fly among the twisting walls of the Arena, scores of deadly laser cannons are firing at you.

If you complete the fifteenth level you get to start over again, with even faster action. Your only reward is survival!! For you see, you are operating under a strict time limit. If you do not complete a level in time, your Huntercraft will disintegrate. HAHAHAHA!! (We told you it was a rather nasty planet.)

INSTRUCTIONS

Type in Listing 1, check it with TYPO



II and SAVE a copy before you RUN it.

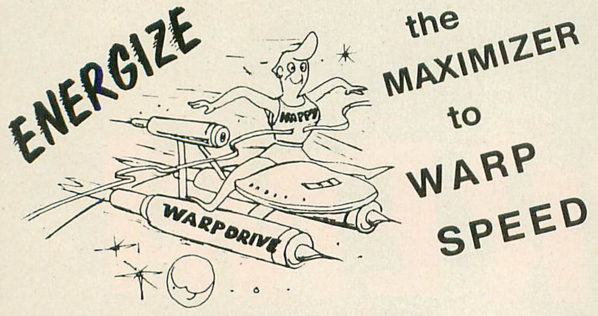
When playing, you start with four Huntercraft and win a bonus ship after every five levels you complete. Maneuver in any direction with the joystick, even diagonally. Pick up an orb simply by moving over it.

Your joystick trigger is the Panic Button. As long as you hold it down, your Huntercraft stays motionless and is invulnerable to laser hits. The good thing about the Panic Button is that it never runs out of operating energy, you can always use it.

continued on page 51

MAXIMIZE STORAGE CAPACITY ON YOUR ATARI 1050* DISK DRIVE WITH THE HAPPY 1050 MAXIMIZER™

Now you can store **twice** as much data on your **ATARI 1050** disk drive with this easy to install high quality plug in adapter. Requires no soldering and no permanent modifications. Runs all popular true double density programs, utilities, and operating systems.



You can upgrade your **HAPPY 1050 MAXIMIZER** to a **WARP SPEED HAPPY 1050 ENHANCEMENT™.** Improves reading and writing speed 500% and comes with the **HAPPY COMPUTERS WARP SPEED SOFTWARE™** package. Makes your **ATARI 1050** the most powerful disk drive available. Easy plug in installation lets you upgrade your **HAPPY 1050 MAXIMIZER** to **WARP SPEED at any time.**

Take COMMAND with the HAPPY 1050 CONTROLLER™

When used with the **ENHANCEMENT** or **MAXIMIZER** allows writing on the flip side of disks without punching holes. Selects protection from writing on valuable disks. Selection can be made both from software commands and a three position switch. When used with the **ENHANCEMENT** allows both switch and software control of reading and writing speeds. Plug in installation requires no soldering. May be used without **ENHANCEMENT** or **MAXIMIZER** with manual control of write protection.

Discount prices through Dec. 31, 1984:

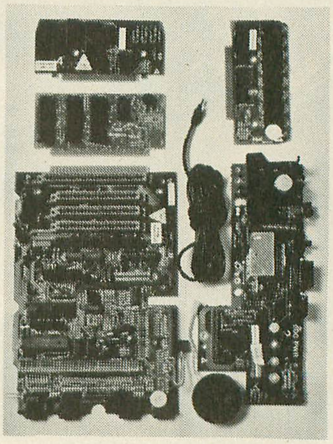
HAPPY 1050 MAXIMIZER complete $124.95
MAXIMIZER to **ENHANCEMENT UPGRADE** $129.95
(You must already have a Happy 1050 Maximizer)
HAPPY 1050 MAXIMIZER with factory installed **MAXIMIZER** to **ENHANCEMENT** upgrade, same as **WARP SPEED HAPPY 1050 ENHANCEMENT** $249.95
HAPPY 1050 CONTROLLER $49.95
WARP SPEED HAPPY 810 ENHANCEMENT™ for 810 disk drive (supports high speed single density) $249.95

Price above include free delivery in the USA. California residents add 6.5% sales tax.

*Note: ATARI 1050 is a trademark of Atari, Inc.

HAPPY COMPUTERS, INC.
P.O. Box 1268, Morgan Hill, CA 95037
(408) 779-3830

SPARE PARTS FOR YOUR ATARI



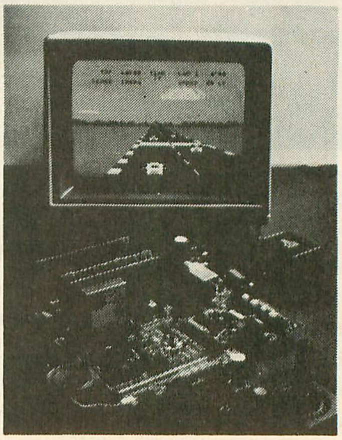
New Spare Parts For Atari 800/400/810

Replacement Printed Circuit Boards (PCB) w/parts

800 Main $30	400 Main $20	810 side $50
800 Power $5	400 Power $4	810 side w/DS...$75
CPU w/GTIA.$20	16K RAM $25	810 Analog ... $20
10K OS $15		810 Power $25

Hard to find Integrated Circuits $5. each
On CPU: GTIA, ANTIC, CTIA, CPU 6502, CPU 6511
On 10K OS: Math ROM 399B, OS ROMs 499B & 599B
On 800/400 Main: Pokey, 6520 PIA
ON 810 & 850: MPU 6507, PIA 6532, RAM 6810, ROM C
Power Paks 800/810 .... $20 ea 800 XL .... $30 ea
Limited quantity used 800 cases & cast shields $40 ea

B&C computervisions (408) 554-0666
3400 El Camino Real, #1, Santa Clara, CA 95051
Hours: Tuesday-Friday 10am-7pm/Sat. 10am-5pm
Terms: UPS Shipments free within USA for orders over $50. Add $5 if under $50. COD or prepaid. Calif. Res. add 6½% sales tax.



800 PCB Sets Main, CPU, 10K OS, Power & RAM $75 Less RAM chips $55

810 PCB Set w/side, DS, Power, Analog $100
810 Drive Mechanisms Tandon or MPI $100
Anti Static work pad 18in x 24in $13
Field Service Manuals 800/400, 800XL or 810 $25 ea
For 1050 or 1200XL $20 ea For 410 or 835 $15. ea
Diagnostic Cartridges Computer or Disk $25 ea
Happy Upgrades, 800, 800XL, 810, 1050, 1030 $CALL
Books, Modems, Monitors, Printers, Joysticks $CALL
Aventure International Gold Series $40.
Software by SSI, OSS, Synapse, LJK, Atari ETC.
Atari 800/400 Technical Reference Notes $20.
Pilot, Basic, Microsoft II, Assembler Manuals $5 ea

SPARE PARTS FOR YOUR ATARI

ARENA RACER
continued from page 49

The bad thing about the Panic Button is that it makes your time limit run out faster. . . Is Arena a hard game to play? All I can say is that I wrote it and I've never gotten higher than 12 of the 15 levels.

PROGRAM TAKE-APART

Initial setup is done in Lines 1000 to 1070. This includes POKEing in the machine language subroutine and character definition.

The major action is found in lines 10 to 70:

10	Produces timer sound and checks for OUT OF TIME
15	Draws current screen and checks for PANIC BUTTON
20	Shuts off timer sound and checks JOYSTICK
30	BOOLEAN LOGIC determines joystick dirction
40	Checks to see if you hit something
50–60	If path is clear, lets you move in that direction
70	Go back for more

Lines 80 to 280 check what character your ship has hit. The appropriate action is then taken—pick up orb, be destroyed by laser, etc.

CUSTOM SCREEN SETUP

The more adventurous of you can create your own screens. It's not exactly easy, but it can be done after some trial and error.

The Arena is made up of an array of 70 X 70 characters. DATA for the levels is contained in lines 6000 to 6290.

Every second line, beginning at 6000, contains 70 characters arranged in a pattern which makes up the walls, spaces and cannons. Every second line beginning at 6010 contains DATA for placing the four orbs. You may manipulate this DATA as you wish. The only restriction is that first five and the last five characters in each of the screen DATA statements must each be an "A."

The screen DATA characters represent:

A	Solid line of wall characters
B	Solid line of open spaces
C,D,F	3 different patterns of walls and spaces
E,G	2 patterns of walls and spaces that also include laser cannons

Placement of the orbs is more difficult. The four orbs must not be in the path of any laser, on or off the small viewscreen, or they will be destroyed. To determine where you want each orb, multiply the vertical coordinate by 70, add the horizontal coordinate, and place the result in the DATA line immediately following the screen DATA line. This should be done with each of the four orbs for each screen.

Below is an example of a customized first screen. Line 6000 is the screen data, and line 6010 is the placement data for the four orbs.

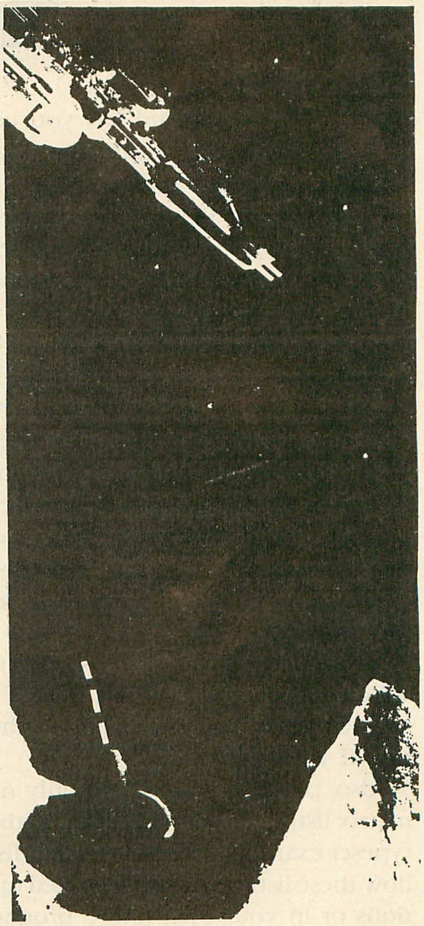
```
6000 DATA AAAAABBBBBBEB
BBBBBBEBBBBBBBEBBBBBBEBB
BBBBBEBBBBBBBEBBBBBBBBEB
BBBBBBAAAAA
6010 DATA 672,1338,2075
,3043
```

The only way to test your Arena and make sure that everything is working right is to RUN the game. If you want to test a higher level, change L=0 in line 1070. L is the current level minus one. So L=7 would start you out at level 8.

Now you should have enough to get started. Have fun creating your own Arenas.

*Jamie Sutherland is a high school junior from Bend, Oregon. His first hands-on computing experience was with the old Sinclair ZX-80 of popular **Antic** game programmer J.D. Casten. You'll see Jamie's vast scrolling maze game, "Valiant," in **Antic** soon.*

Listing on page 76.



HANDY USR ROUTINES

Machine language power from BASIC

by ERNIE NEGUS

A collection of short but powerful USR machine language routines that can be usefully plugged into your your BASIC programs for any Atari computer. Antic Disk subscribers ENTER "D:USERCALL.LST" and follow the instructions in the article.

BASIC's USR command lets you use speedy, memory-efficient machine language subroutines from the comfort of BASIC. Although these subroutines are tricky to type, they let your BASIC programs access a wider variety of functions, run faster and occupy less memory.

Simply put, machine code is a series of numbers stored in memory. Each number stands for an instruction to the computer. Once you have the address of this code, the USR function lets you run it from BASIC.

Machine code can be stored in several ways. Here, the code is stored in character strings. We can now use BASIC's ADR function to find the beginning address of the character string, and put this value into our USR call.

Since the computer interprets each character as a machine code instruction, a single mistyped character could cause your computer to lock up when you RUN the program.

You must type in each character string exactly as it appears, checking each line with TYPO II. Refer to the Special Atari Character charts in the **Antic** listings section if you need additional help.

Also, please note that you only need to type the dot-matrix listing printouts marked with TYPO II codes. The typeset example lines below each listing simply indicate how these listings should be called up in these demonstrations or in your own BASIC programs.

All USR calls have the general form: X=USR(z, p1, p2, p3) where z is the decimal address of the machine language routine, and p1, p2 and p3 are input values (parameters) which the machine language routine will use. While some USR calls do not need parameters, others demand several. Any parameters you use must be numbers between 0 and 65,535.

Here are some short USR routines which use machine language to manipulate bits, perform multiple PEEKs and POKEs, evaluate Boolean expressions, and even play music.

Let's examine our USR routines and see how they work:

```
IZ 10 BBYT=ADR("h*)♥.TaUhhP' 8(├┘&TJPS♦")
```

15 ANS=USR(BBYT,p1,p2,p3,p4,p5,p6,p7,p8)

This routine converts binary numbers to decimal. In our example, the mysterious-looking character string is the machine language subroutine. The ADR function determines the decimal address of the subroutine. This address is stored in the variable BBYT. Our final result will be stored in ANS.

We'll use this routine to convert 00001111 to decimal:

First, type in the machine language string as shown in line 10.

Now, type:

```
15 ANS=USR(BBYT,0,0,0,0,1,1,1,1)
17 PRINT ANS
```

When you RUN this demonstration, ANS will be equal to 15, the decimal equivalent of 00001111.

USR routines which do not return values to your BASIC program, such as MPOK, must also be equated to a BASIC variable. Programmers often call these "dummy" variables because they aren't used in any calculations, but are required by BASIC syntax rules.

```
RB 20 DPEK=ADR("hh.M.h.L ♥1L.TH1L.U◆")
```

```
25 ANS=USR(DPEK,<address>)
```

This routine performs a double PEEK at any given decimal address. Its BASIC equivalent is:

```
ANS=PEEK (<address>)+PEEK (<address+1>) * 256
```

Please note that items within angle brackets, such as <address>, tell you what type of data the USR routine requires. If you wanted to perform a double PEEK at address 1536, for example, you would type:

```
25 ANS=USR(DPEK,1536)
```

```
WE 30 MPOK=ADR("h.h.Mh.L ♥hh.LHJP.◆")
```

```
35 DUMMY=USR(MPOK,<address>,<byte1>,
<byte2>...)
```

MPOK will POKE any number of bytes into successive memory locations, beginning at the specified decimal address.

```
GW 40 DPOK=ADR("h.Jh.Uh.T ♥h.Uh.TH%V.THJP
.◆")
```

```
45 DUMMY=USR(DPOK,<address>,<word1>,
<word2>...)
```

DPOK works just like MPOK, except DPOK will POKE any number of words into successive memory locations.

```
DG 50 HI=ADR("hh.Th.♥.U◆")
```

```
55 ANS=USR(HI,<word>)
```

HI returns the high byte of any word. Its BASIC equivalent is:ANS=INT(WORD/256)

```
RN 60 LOW=ADR("hhh.T.♥.U◆")
```

```
65 ANS=USR(LOW,<word>)
```

LOW returns the low byte of any word. Its BASIC equivalent is:

```
ANS=WORD-INT(WORD/256)*256
```

```
AA 70 BAND=ADR("h..T.Uh%U.Uh%T.TJPS◆")
```

```
75 ANS=USR(BAND,p1,p2,p3...)
```

BAND performs a logical "AND" on the bits of any number of parameters. BAND can be used to separate missiles and test the direction bits of the joystick ports.

```
FU 80 BOR=ADR("h.♥.T.Uh.U.Uh.T.TJPS◆")
```

```
85 ANS=USR(BOR,p1,p2,p3...)
```

BOR performs a logical "OR" on the bits of any number of parameters. BOR can be used to set bits and alter display list options.

```
IO 90 BXOR=ADR("h.♥.T.UhEU.UhET.TJPS◆")
```

```
95 ANS=USR(BXOR,p1,p2,p3...)
```

BXOR performs a logical "Exclusive OR" on the bits of any number of parameters. The routine can be used to control blinking characters and flashing colors.

```
LO 100 BROT=ADR("hh.h.L◆.N.Khh.n.K
hh.8 K♥♥.U.h.T◆")
```

```
105 ANS=USR(BROT,<address>,<direction>,<carry>)
```

BROT will rotate the bits of a byte in RAM. In line 75, <address> is the decimal address of this byte, <direction>, is the direction of rotation (use 0 for right rotation, 1 for left rotation), and <carry> initializes the carry bit. ANS will contain the carry condition after the rotation.

BROT can be used to rotate characters and players, and convert decimal numbers to binary numbers.

```
LJ 110 RORB=ADR("h./h K♥Py1P.T1N.U.P%Pe
L.P%QeM.Q1P.U%T.P%U.T%PENPa%QEQP.◆")
```

continued on next page

the toolbox

115 DUMMY=USR(RORB,<start>,<end>,<skip>)

```
RJ 120 ROLB=ADR("h█/h K♥Py1N┐T1P┐U┌N8%Ne
   L┐N%0eM┐01N┐U%T┌N%U┐T%PENPa%QEOP[◆")
```

125 DUMMY=USR(ROLB,<start>,<end>,<skip>)

RORB and ROLB will rotate bytes from decimal address <start> to address <end>, while skipping every <skip> bytes. Of course, the difference between <start> and <end> should be evenly divisible by <skip>. Otherwise, your program may lock up.

These routines can be used for coarse scrolling, animating characters, moving players and missiles vertically, and changing display lists.

```
WP 130 MUSIC=ADR("hh.Lh.Khh.M ♥}┘▬ R)♥▬◢R
   ▬ R┐●ǁ d●P) ▬ ├R)(▬┘R1K▬ RH1K*)♥┐●d●P▮
   HFMPK◆")
JX 131 DIM MUSDAT$(26)
CA 132 MUSDAT$="<▪D▪L▪D▪<▪<▪<▪<▪D▪D▪<▪D▪L
      ←":AD=ADR(MUSDAT$)
CE 133 LN=LEN(MUSDAT$)/2
UK 134 DUMMY=USR(MUSIC,AD,LN)
```

This routine lets the computer play simple tunes from data stored in MUSDAT$. Each note in MUSDAT$ is represented by two bytes. The first byte of each pair is the pitch value of the note. Consult your BASIC reference manual for appropriate pitch values. The second byte is the duration of the note, in jiffies.

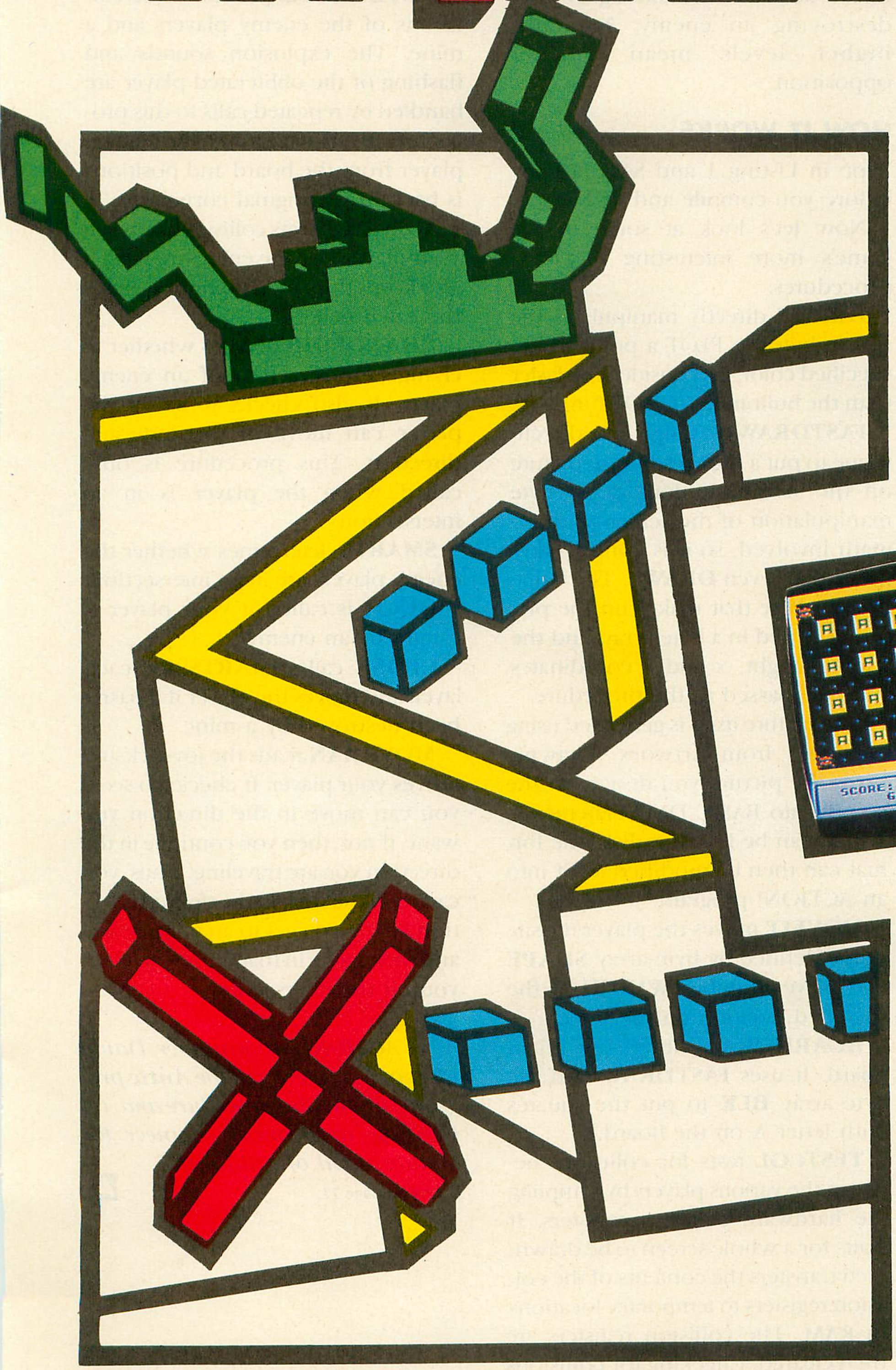
Whole notes require approximately 60 jiffies, quarter notes use approximately 15 jiffies. In our example, MUSDAT$ holds the data for the last two bars of "Mary Had A Little Lamb."

USR routines are easy to use and can breathe new life into tired BASIC programs.

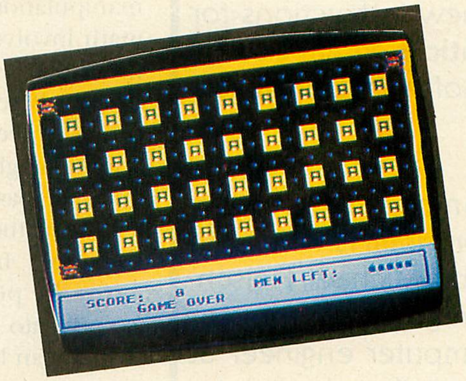
Ernie Negus is a computer engineer for Intel in Oregon, working mainly on state-of-the-art hard disks, 32-bit microprocessors and quad density floppies.

AMAZING

by DAVID PLOTKIN



A challenging maze chase game that demonstrates the speed and versatility of the ACTION! language. Requires ACTION! cartridge from Optimized Systems Software. Works on all Atari memory computers with 32K disk or 24K cassette. Antic Disk subscribers will find a "run-time" version on their disk, for playing without the cartidge.



Amazing is a surprisingly imaginative maze chase game written in ACTION! You are a skinny red X named Gork. All you want from life is to wander the city grid, munching up the energy pellets that the programmer thoughtfully left strewn about.

Not surprisingly, three enemies will attempt to stop you with their instantly lethal touch. Luckily, your unique defensive mines can immobilize and vaporize enemies. But of course each opponent is quickly replaced by another.

continued on next page

NEXT ISSUE

IMPROVED PRINTED LISTINGS

Spaces between Atari special characters will make **Antic** program listings easier than ever to type correctly.

See the new instructions for Typing **Antic** Program Listings in June's Software Library section.

Antic's improved custom printing program is written in ACTION! by Michael Fleischmann, a regular contributor and a computer engineer at Hill Air Force Bace in Utah.

NEXT ISSUE

Release a mine by pressing the joystick button. You can have up to four mines on the board at one time. To retrieve an unused mine, touch it. The mines become available again after destroying an enemy. Naturally, higher levels mean tougher opposition.

HOW IT WORKS

Type in Listing 1 and SAVE a copy before you compile and RUN it.

Now let's look at some of the game's more interesting ACTION! procedures.

DRAW7 directly manipulates the screen bytes to PLOT a point in the specified color. It's considerably faster than the built-in Atari PLOT function.

FASTDRAW is a high speed technique to put a high resolution picture on the screen. It does direct byte manipulation of the screen with no math involved, so it is considerably faster than even **DRAW7**. The value of each byte that makes up the picture is stored in a byte array, and the width, height, x and y coordinates must be passed to the procedure.

The picture itself is generated using **Drawpic** from Artworx. Drawpic turns the picture you design on the screen into BASIC DATA statements, which can be listed to disk; the format can then be modified to fit into an ACTION! program.

MOVEIT moves the player/missile shape defined by byte array **SHAPE** and player number **WHICH** to the specified position on the screen.

BOARDDRAW draws the initial board. It uses **FASTDRAW** and the byte array **BLK** to put the squares with letter A on the board.

TESTCOL tests for collisions between the various players by sampling the hardware collision registers. It waits for a whole screen to be drawn, then transfers the contents of the collision registers to temporary locations in RAM. The collision registers are then cleared. Checking for collisions is actually done by looking at the temporary locations.

LLOC performs the same function as LOCATE, but much faster.

GOTBUMPED processes the collisions of the enemy players and a mine. The explosion sounds and flashing of the obliterated player are handled by repeated calls to this procedure. It also removes the enemy player from the board and positions is back in its original corner.

MUNCH detects collisions between your player and the energy pellets. It also keeps the sound going and erases the eaten pellet.

CHANGEDIR decides whether to change the direction of an enemy player. It also checks to see if the player can move in the indicated direction. This procedure is only called when the player is in an intersection.

SMARTS determines whether the enemy players are in an intersection.

OUCH is called if your player is caught by an enemy.

CHASE calls **SMARTS** for each layer, and moves the player if it hasn't been destroyed by a mine.

MOVEMAN reads the joystick and moves your player. It checks to see if you can move in the direction you want. If not, then you continue in the direction you are traveling. Thus, you can push the stick in the desired direction *before* you get to an intersection and then move in that direction when you hit the intersection.

*Avid ACTION! programmer David Plotkin is a veteran of the **Antic** program submission procedure and, on the side, a chemical engineer for Standard Oil of California.*

Listing on page 77.

FADER II

Enhanced dot-by-dot picture dissolves!

by PATRICK L. DELL'ERA

Last month, we published Patrick Dell'Era's reworking of Philip Price's "Picture Painter." We now present Patrick's Fader II, based upon the popular dot-by-dot picture dissolve program by Joseph Grande that **Antic** printed in the September 1984 graphics issue.

The original Fader was written in the BASM language, which is now often hard to find. Patrick has disassembled the object code into full MAC/65 source code, and re-written it with several improvements including a special modification program in BASIC. —ANTIC ED

This is an enhancement of the popular "Fader" program that appeared in the September, 1984 ***Antic****. The original program created a dot-by-dot "lapse-dissolve" effect on Micro Painter files. Fader II now works on Micro Illustrator pictures as well. The program is complete in this issue—including source code. It will work on all Atari computers with 48K and disk drive.*

THE LISTINGS

There are three listings. Listing 1, FADERII.BAS, is the main program. It is written in BASIC and creates a machine language binary file on your disk called AUTORUN.SYS.

Listing 2, FADERMOD.BAS, is a BASIC program which can be used to modify certain of the elements in AUTORUN.SYS. Listing 3, FADERII.ASM, is the assembly language source code and need not be typed in. It is included for machine language programmers who may wish to study and further modify the program.

Type in Listing 1 and check it especially carefully with TYPO II. Make sure you haven't skipped any lines. TYPO II can't check for that.

SAVE a copy of the program. When you RUN Fader II, it will read all of those DATA statements (they are the machine code) and then prompt you to ready your disk and press [RETURN].

Place a fresh, formatted disk in your drive containing DOS 2.0S and two or more picture files. Press [RETURN], and a file named AUTORUN.SYS will be written to your disk. Now, boot the disk and the slide show will begin.

SLIDE SHOW II

Unlike the original program, Fader II can handle both uncompressed and compressed Graphics 7+ picture files. This means you can have any combination of Micro Painter or Micro Illustrator files on the same disk. (Micro Illustrator software comes with the KoalaPad, the Atari Touch Tablet and other widely used graphics products.)

So that Fader II can tell the difference between the file types, you must use a .PIC extender on your compressed files. The Micro Painter files should have a .?IC extender (where ? designates any number or letter other than P).

Fader II will cycle through each picture creating a screen pixel dissolve. When it reaches the last file, it will begin again with the first. As with the original program, you may skip the pause between pictures by pressing [START].

In Fader II, after a picture fades in, it may be held indefinitely on the screen by pressing [OPTION]. In this way, one may take a good look at a particularly fetching piece of art, or gracefully change the disk without racing the clock. If you have DOS on your disk, pressing [SELECT] will take you to it.

Fader II does not sit in the DUP.SYS area of memory, so a Binary Load can

continued on next page

be done on it from DOS without creating a MEM.SAV file. It will run with or without a cartridge installed.

FADER II MODIFICATION

Although Fader II has an automatic pausing rate, the modification program, FADERMOD.BAS, will allow you to change the length of time the picture is left on the screen—as well as the drive that the pictures are loaded from. This allows quite an effective display for Ramdisk owners who load their pictures into the simulated 128K disk and designate it as the load drive.

Type in Listing 2, again checking it with TYPO II and SAVEing a copy. When RUN, it will ask for the file name (i.e. AUTORUN.SYS, FADERII.EXE, etc.). It then uses NOTE and POINT to read the appropriate variables. Each time the [OPTION] key is pressed, the drive number is incremented by 1. If it reaches 5, it rolls back to 1.

The pause rate is modified by pressing [SELECT]. Each time [SELECT] is pressed, the pause time is increased by 4.27 seconds until it reaches (255 * 4.27). It then becomes 0. If the minus key [–] is pressed, the pause time will be reduced with each press of [SELECT]. Pressing any other key puts it back into the increment mode.

When your variables are set to your satisfaction, press [START] to save the changes to your program. It is important that the same disk Fader II is read from should be the disk to which the changes are written. Otherwise, an innocent bystander on another disk could be modified, surely causing its demise.

LISTING 3

As mentioned above, Listing 3 is the MAC/65 assembly source code. It was created with the MAC/65 Assembler Editor by Optimized Systems Software, Ultra Disassembler by Adventure International, and OmniMon by CDY Consulting. Without these three excellent programming tools, the above code could not have been recreated and modified.

Patrick L. Dell'Era is a member of the San Francisco Atari users' group ABACUS, and a strong assembly language programmer whose time is currently being taken up with a new baby.

Listing on page 70.

SOFTWARE LIBRARY

Antic type-in listing section includes every full-length program from this issue. Listings are easier to type and proofread, easy to remove and save in a binder if you wish.

DISK SUBSCRIBERS: You can use all these programs immediately. Just follow the instructions in the accompanying magazine articles.

Antic program listings are typeset by Star's Gemini 10X Printer—From Star Micronics, Inc., 200 Park Avenue, New York, NY 10166.

TYPING SPECIAL ATARI CHARACTERS

Shown below are the Atari Special Characters as printed in **Antic** listings—and the keys you must type in order to get them. Boxes are drawn around the normal video characters here so you can see their positions more accurately, these boxes do not appear in the printed listings.

Whenever the CTRL key (CONTROL on XL models) or SHIFT key is used, *hold it down* while you press the next keys. Whenever the ESC key is used, *press and release* it before typing the next keys.

Turn on inverse video by pressing the Atari logo key [Atari logo] once. Turn it off by pressing a second time. (XL models use the Reverse Video Mode Key [Reverse Video key] instead.)

Sometimes it's not easy to tell apart the following characters, shown here in both normal and inverse video. Be especially careful when you type any of these:

	CTRL F		/
	CTRL G		SHIFT +
	CTRL N		SHIFT -
	CTRL R		-
	CTRL S		+

INVERSE VIDEO

FOR THIS	TYPE THIS	FOR THIS	TYPE THIS
	[Atari logo] CTRL ,		[Atari logo] CTRL Y
	[Atari logo] CTRL A		[Atari logo] CTRL Z
	[Atari logo] CTRL B		ESC SHIFT DELETE
	[Atari logo] CTRL C		
	[Atari logo] CTRL D		
	[Atari logo] CTRL E		ESC SHIFT INSERT
	[Atari logo] CTRL F		
	[Atari logo] CTRL G		
	[Atari logo] CTRL H		ESC CTRL TAB
	[Atari logo] CTRL I		
	[Atari logo] CTRL J		
	[Atari logo] CTRL K		ESC SHIFT TAB
	[Atari logo] CTRL L		
	[Atari logo] CTRL M		
	[Atari logo] CTRL N		[Atari logo] CTRL .
	[Atari logo] CTRL O		[Atari logo] CTRL ;
	[Atari logo] CTRL P		[Atari logo] SHIFT =
	[Atari logo] CTRL Q		ESC CTRL 2
	[Atari logo] CTRL R		ESC CTRL DELETE
	[Atari logo] CTRL S		
	[Atari logo] CTRL T		
	[Atari logo] CTRL U		ESC CTRL INSERT
	[Atari logo] CTRL V		
	[Atari logo] CTRL W		
	[Atari logo] CTRL X		

NORMAL VIDEO

FOR THIS	TYPE THIS	FOR THIS	TYPE THIS
	CTRL ,		CTRL T
	CTRL A		CTRL U
	CTRL B		CTRL V
	CTRL C		CTRL W
	CTRL D		CTRL X
	CTRL E		CTRL Y
	CTRL F		CTRL Z
	CTRL G		ESC ESC
	CTRL H		ESC CTRL -
	CTRL I		ESC CTRL =
	CTRL J		ESC CTRL +
	CTRL K		ESC CTRL *
	CTRL L		CTRL .
	CTRL M		CTRL ;
	CTRL N		SHIFT =
	CTRL O		ESC SHIFT CLEAR
	CTRL P		
	CTRL Q		
	CTRL R		ESC DELETE
	CTRL S		ESC TAB

HOW TO USE TYPO II

Type in TYPO II and SAVE a copy to disk or cassette.

Type GOTO 32000 and follow TYPO II onscreen instructions. If the resulting two-letter line codes are not exactly the same as those in the magazine, you mistyped something in that line.

To call back any line previously typed, type an asterisk [*] followed (without in-between spaces) by the line number, then press [RETURN]. When the complete line appears at the top of the screen, press [RETURN] again. This is also the way you use TYPO II to proofread itself.

To LIST your program, press [BREAK] and type LIST. To return to TYPO II, type GOTO 32000.

To remove TYPO II from your program, type LIST "D:FILENAME",0,31999 [RETURN] (Cassette owners LIST "C:). Type NEW, then ENTER "D:FILENAME" [RETURN] (Cassette—ENTER "C:). Your program is now in memory without TYPO II and you can SAVE or LIST it to disk or cassette.

Owners of the BASIC XL cartridge from O.S.S. type SET 5,0 and SET 12,0 before using TYPO II.

```
WB 32000 REM TYPO II BY ANDY BARTON
VM 32010 REM VER. 1.0 FOR ANTIC MAGAZINE
HS 32020 CLR :DIM LINE$(120):CLOSE #2:CLO
   SE #3
BN 32030 OPEN #2,4,0,"E":OPEN #3,5,0,"E"
YC 32040 ? "K":POSITION 11,1:? "TYPO II"
EM 32050 TRAP 32040:POSITION 2,3:? "Type
   in a program line"
HS 32060 POSITION 1,4:? " ":INPUT #2;LINE
   $:IF LINE$="" THEN POSITION 2,4:LIST B
   :GOTO 32060
XH 32070 IF LINE$(1,1)="*" THEN B=VAL(LIN
   E$(2,LEN(LINE$))):POSITION 2,4:LIST B:
   GOTO 32060
TH 32080 POSITION 2,10:? "CONT"
MF 32090 B=VAL(LINE$):POSITION 1,3:? " ";
NY 32100 POKE 842,13:STOP
CN 32110 POKE 842,12
ET 32120 ? "K":POSITION 11,1:? " TYPO II 
   ":POSITION 2,15:LIST B
CE 32130 C=0:ANS=C
QR 32140 POSITION 2,16:INPUT #3;LINE$:IF
   LINE$="" THEN ? "LINE ";B;" DELETED":G
   OTO 32050
VV 32150 FOR D=1 TO LEN(LINE$):C=C+1:ANS=
   ANS+(C*ASC(LINE$(D,D))):NEXT D
WJ 32160 CODE=INT(ANS/676)
JW 32170 CODE=ANS-(CODE*676)
EH 32180 HCODE=INT(CODE/26)
BH 32190 LCODE=CODE-(HCODE*26)+65
HB 32200 HCODE=HCODE+65
IE 32210 POSITION 0,16:? CHR$(HCODE);CHR$
   (LCODE)
VG 32220 POSITION 2,13:? "If CODE does no
   t match press RETURN and edit line a
   bove.":GOTO 32050
```

ERROR FILE

BUS OVERLINES

Some signals and address labels were printed without overlines in Part III of Earl Rice's Parallel Bus Revealed" (Antic, March 1985).

These are the correct labels:

$\overline{\text{D8XX - DFXX}}$
$\overline{\text{CS}}$ (CHIP RESET)
R/$\overline{\text{W}}$
$\overline{\text{DIXX}}$
$\overline{\text{RDE}}$ (READ DATA ENABLE)
$\overline{\text{DS}}$ (DATA STROBE)
$\overline{\text{DRST}}$ (DEVICE RESET)

FIRST LESSON IN ASSEMBLY

Line 100 of the listing for "First Lesson in Assembly Language" (November, 1984) should read POKE 755,4 instead of POKE 775,4.

KOOKY'S QUEST

February '85

The following line is missing:
2100 FOR S=32 TO 16 STEP −4: SOUND 0,S,14,10: EA=EA *EA*EA: SOUND 0,0,0,0: EA=1 ∧0:NEXT S

DRUM SYNTH

February '85

In Figure 1, the "ART" should be the Fuji (inverse) symbol.

MISSING INFOBITS

DECEMBER '84

The AL source listing for Infobits (Dec. '84) was left out of the previous issue. You'll find it in the Jan. '85 Software Library.

ADVENT X-5

November '84

Missing line: 8020 RUN. Also, cassette owners should change the 138 in line 4005 to 130. The TYPO II code for line 1005 is EJ.

ADVENTURE ISLAND

November '84

Line 837 is missing its last item of data, a 4. Also, it will not run with DOS XL.

BEER PARTY ATARI

Article on page 43.

LISTING 1

```
GR 10 REM BEER PARTY ATARI
TR 20 REM BY DR. JOHN FERGUSON
RH 30 REM ANTIC MAGAZINE
VC 50 DIM S$(10)
TR 100 CLOSE #1:OPEN #1,4,0,"K:":REM OPEN
   KEYBOARD FOR INPUT
IV 110 GRAPHICS 0:? :? :? "DO YOU WANT TO
   ADD TO A PREVIOUS FILE":? :? :? "(Y/N
   )? ";
NM 120 GET #1,K:IF K<>89 THEN BA=0:BB=0:B
   C=0:BD=0:BE=0:N=0:GOTO 1000
PU 130 ? :? :? :? :? "MAKE SURE 'BEERDATA
   ' IS IN DRIVE AND  PRESS ANY KEY.":? :
   ? :?
GP 140 GET #1,K
KL 150 CLOSE #2:OPEN #2,4,0,"D:BEERDATA"
WT 160 INPUT #2;BA,BB,BC,BD,BE,NUM:CLOSE
   #2:REM RETRIEVE OLD DATA
UG 1000 GRAPHICS 8:REM CLEAR MEMORY
GV 1010 GOSUB 10010:REM CHANGE CHARACTER
   SET
LW 1015 GRAPHICS 0:POKE 709,136:POKE 710,
   2:POKE 712,2:POKE 752,1:REM SET COLORS
    AND BLANK CURSOR
NW 1020 POKE 756,CHSET/256
HP 1030 DL=PEEK(560)+256*PEEK(561):REM CU
   STOMIZE DISPLAY LIST
YB 1040 FOR N=6 TO 9:POKE DL+N,7:NEXT N
SP 1050 POKE DL+10,6:POKE DL+11,8:POKE DL
   +3,66+7:POKE DL+21,6:POKE DL+22,6
II 1055 POKE DL+27,65:POKE DL+28,PEEK(560
   ):POKE DL+29,PEEK(561)
UH 1057 POKE 82,0:POKE 82,0:POSITION 0,0:REM SET FI
   RST SCREEN
VI 1060 POKE 87,2:? #6,"                beer
        #$             '*    %&  sampling
    []"
LC 1070 ? #6," rate beers  1 to 9"
TQ 1080 POKE 87,0:POSITION 4,4
UM 1110 ? "Poor               Average       F
   ine"
RA 1120 ? "       ┌────┬────┬────┬────┬────┬────┬
   ──┬─┘"
VP 1130 ? "    1    2    3    4    5    6    7
     8    9":?
TW 1140 ? "         (Enter +5+ if not tast
   ed)"
SC 1150 POKE 85,2:POSITION 14,12:? #6,"BE
   ER +A+ - ?"
XP 1160 ? :? :? :? "      (Push spacebar t
   o see results)"
YG 1170 GOSUB 2010
BC 1200 GOSUB 2110:A=K-48:POSITION 30,12:
   ? "B":GOSUB 2010
DM 1210 GOSUB 2110:B=K-48:POSITION 30,12:
   ? "C":GOSUB 2010
FW 1220 GOSUB 2110:C=K-48:POSITION 30,12:
   ? "D":GOSUB 2010
IG 1230 GOSUB 2110:D=K-48:POSITION 30,12:
   ? "E":GOSUB 2010
OI 1240 GOSUB 2110:E=K-48
NB 1300 REM *  SET SECOND SCREEN
HP 1305 ? "K":POSITION 0,0
BH 1310 POKE 87,2:? #6,"        sampled
        #$             '*    %&    beers
    []"
SE 1315 ? #6," YOUR RATINGS ARE:"
OH 1320 POKE 87,0:POSITION 0,3
KM 1330 ? "              BEER +A+  =  ";A;"
             ":?
NA 1340 ? "              BEER +B+  =  ";B;"
             ":?
PR 1360 ? "              BEER +C+  =  ";C;"
             ":?
SI 1380 ? "              BEER +D+  =  ";D;"
             ":?
TX 1400 ? "              BEER +E+  =  ";E;"
             ":?
DN 1410 POKE 85,2:POSITION 11,12:? #6,"
   CORRECT? (Y:N)"
MS 1430 GET #1,K:IF K<>89 THEN 1015
UN 1440 BA=BA+A:BB=BB+B:BC=BC+C:BD=BD+D:B
   E=BE+E:NUM=NUM+1
YM 1450 CLOSE #2:OPEN #2,8,0,"D:BEERDATA"
   :? #2,BA:? #2,BB:? #2,BC:? #2,BD:? #2,
   BE:? #2,NUM:CLOSE #2:REM SAVE DATA
YB 1500 REM SET THIRD SCREEN
KP 1510 POKE DL+21,2:POKE DL+22,2:? "K":P
   OSITION 0,0
SC 1520 POKE 87,2:? #6,"          taste
        #$             '*    %&  results
    []"
TA 1530 ? #6,"         N = ";NUM
YH 1540 POKE 87,0:POKE 82,0:POSITION 0,3
NG 1550 ? "BUSCH          ┤";:L=BB:GOSUB 22
   05:REM BEER B
SJ 1560 ? "               │"
FT 1570 ? "BLACK LABEL    ┤";:L=BD:GOSUB 22
   05:REM BEER D
SP 1580 ? "               │"
WT 1590 ? "MILLER         ┤";:L=BE:GOSUB 22
   05:REM BEER E
RT 1600 ? "               │"
FC 1610 ? "MEISTER BRAU   ┤";:L=BA:GOSUB 22
   05:REM BEER A
RZ 1620 ? "               │"
FI 1630 ? "PABST LIGHT    ┤";:L=BC:GOSUB 22
   05:REM BEER C
DU 1640 ? "               └──┬──┬──┬──┬──┬──
   ──┬──┬──┘"
LL 1650 ? "                1  2  3  4  5  6
    7  8  9"
ZW 1670 ? "                    Average Sco
   re"
PB 1680 ? :? :? "     (Press spacebar to e
   nter data)"
HN 1700 GET #1,K:IF K<>83 THEN 1015
VJ 1800 REM SHOW SCORE ROUTINE
IO 1810 TRAP 40000:TRAP 1510
TR 1820 L=BB:POSITION 33,3:GOSUB 2310
WG 1830 L=BD:POSITION 33,5:GOSUB 2310
YM 1840 L=BE:POSITION 33,7:GOSUB 2310
YZ 1850 L=BA:POSITION 33,9:GOSUB 2310
LU 1860 L=BC:POSITION 33,11:GOSUB 2310
RF 1870 GOTO 1700
AD 2000 REM GET KEY SUBROUTINE
JU 2010 GET #1,K:IF K=32 THEN POP :GOTO 1
   500
KW 2020 IF K<49 OR K>57 THEN ? CHR$(253):
   GOTO 2010
AJ 2030 RETURN
ZO 2100 REM ENTER DATA SUBROUTINE
VU 2110 POSITION 35,12:? K-48:SOUND 0,40,
   10,14:FOR T=1 TO 50:NEXT T:SOUND 0,0,0
   ,0:POSITION 35,12:? "?":RETURN
```

```
OT 2200 REM GRAPHING SUBROUTINE
CQ 2205 IF NUM=0 THEN ? :RETURN
AF 2210 LE=INT((L/NUM)*3)
QY 2220 IF LE=3 THEN ? "!":RETURN
EV 2230 FOR N=4 TO LE:? "!";:NEXT N:? "!"
   :RETURN
OA 2300 REM SCORE FORMAT SUBROUTINE
GO 2310 ? "(";
RV 2320 LE=(L/NUM)+5.0E-03
ZF 2330 S$=STR$(LE)
EA 2340 IF LEN(S$)>3 THEN ? S$(1,4);:GOTO
    2390
HB 2350 IF LEN(S$)=3 THEN ? S$;"0";
XA 2360 IF LEN(S$)=1 THEN ? S$;".00";
PM 2390 ? ")":RETURN
ZY 10000 REM CHARACTER SET SUBROUTINE
GJ 10010 POKE 106,PEEK(106)-5:GRAPHICS 0:
   CHSET=(PEEK(106)+1)*256:? :? "ONE MOME
   NT; ALTERING CHARACTER SET"
MB 10015 CHI=CHSET/256:CLO=0:POKE 203,CLO
   :POKE 204,CHI
AM 10020 DIM XFR$(28):RESTORE 10030:FOR N
   =1 TO 28:READ ML:XFR$(N,N)=CHR$(ML):NE
   XT N
BS 10030 DATA 104,169,0,133,205,168,169,2
   24,133,206,177,205,145,203,200,208
DA 10040 DATA 249,230,204,230,206,165,206
   ,201,228,208,239,96
RK 10050 XFR=USR(ADR(XFR$))
FO 10060 RESTORE 10100
FS 10070 READ A:IF A=-1 THEN RETURN
QL 10080 FOR Z=0 TO 7:READ J:POKE CHSET+A
   *8+Z,J:NEXT Z
AR 10090 GOTO 10070
CN 10100 DATA 1,124,124,56,56,56,16,16,56
DU 10110 DATA 3,127,64,64,64,64,64,127,12
   7
KF 10120 DATA 4,224,32,62,35,33,33,225,22
   5
MV 10130 DATA 5,127,127,127,127,127,127,1
   27,255
QN 10140 DATA 6,225,254,224,224,224,224,2
   24,240
YH 10150 DATA 7,31,16,16,8,15,15,15,7
JH 10160 DATA 10,240,16,16,32,224,224,224
   ,192
YX 10170 DATA 11,36,36,36,0,0,0,0,0
YJ 10180 DATA 59,7,7,3,3,3,1,1,31
EE 10190 DATA 61,192,192,128,128,128,0,0,
   240
EO 10200 DATA -1
```

multiple graphics modes on one screen

MODE MIXER 1

Article on page 38.

LISTING 1

```
AZ 1 REM MODE MIXER
SX 2 REM BY KARL E. WIEGERS
QK 3 REM ANTIC MAGAZINE
VE 10 DIM INC(16),MODE(16),LINES(16),SLPE
   R(15),BPER(15),BAS(15)
JT 12 DIM B$(20),FNAME$(14),A$(100),BL$(3
   9)
RJ 15 BL$(1)=" ":BL$(39)=BL$:BL$(2)=BL$
IL 20 FOR I=2 TO 15:READ A:SLPER(I)=A:NEX
   T I:DATA 8,10,8,16,8,16,8,4,4,2,1,2,1,
   1
WM 25 FOR I=2 TO 15:READ A:BPER(I)=A:NEXT
    I:DATA 40,40,40,40,20,20,10,10,20,20,
   20,40,40,40
MO 30 FOR I=2 TO 15:READ A:BAS(I)=A:NEXT
   I:DATA 0,0,0,0,1,2,3,4,5,6,7,7,8,8
XM 39 REM Display List Interrupt Routine
IB 40 FOR I=0 TO 19:READ A:POKE 1771+I,A:
   NEXT I
HW 50 DATA 72,138,72,169,0,162,28,141,10,
   212,141
ZZ 60 DATA 23,208,142,24,208,104,170,104,
   64
QM 70 OFF=40000:NO=500:MEMORY=550:LOCS=60
   0:SPACE=625
BQ 80 OPEN #1,4,0,"K:"
FA 99 REM Menu
LH 100 GRAPHICS 0:POKE 752,1:POKE 559,0:P
   OKE 710,0:POKE 709,90:POKE 711,198:GOS
   UB LOCS
GN 110 RESTORE 120:FOR I=1 TO 25:READ A:P
   OKE DL+I+2,A:NEXT I
OV 120 DATA 71,0,0,7,6,6,2,2,2,2,2,2,2,2,
   2,2,2,2,2,2,6,6,6,6,65
CE 130 POKE DL+4,LMEM:POKE DL+5,HMEM:POKE
    559,34
LC 135 POKE 87,2
PI 140 POSITION 5,0:? #6;"mode MIXER"
KK 145 POKE 87,1:BYTE=40:GOSUB MEMORY
PH 150 POSITION 1,0:? #6;"PICK AN OPTION:
   "
WV 155 POKE 87,0:BYTE=40:GOSUB MEMORY:POK
   E 82,6
JA 160 POSITION 6,1:? "1 - CREATE A NEW D
   ISPLAY"
QC 161 ? :? "2 - CHANGE THE CURRENT DISPL
   AY"
TJ 162 ? :? "3 - SEE THE CURRENT DISPLAY"
XH 163 ? :? "4 - SAVE CURRENT DISPLAY ON
   DISK"
GD 164 ? :? "5 - LOAD A DISPLAY FROM DISK
   "
ZA 165 ? :? "6 - DONE WITH THIS PROGRAM"
IS 170 POKE 82,2
PH 175 POKE 87,1:BYTE=560:GOSUB MEMORY
SN 180 POSITION 7,0:? #6;"?"
AJ 190 GET #1,CH:CH=CH-48
XV 200 POSITION 9,0:? #6;CH
NT 210 IF CH<1 OR CH>6 THEN GOSUB NO:? #6
   :? #6;"  enter only 1-6";:GOTO 190
ZR 220 ON CH GOTO 1000,2000,6000,7000,300
   0
KN 230 GRAPHICS 0:END
UM 500 FOR I=1 TO 15:SOUND 0,96,10,10:NEX
   T I
VU 510 FOR I=1 TO 15:SOUND 0,128,10,10:NE
   XT I
RF 520 SOUND 0,0,0,0:RETURN
BX 550 HMEM=256*HMEM+LMEM+BYTE:LMEM=HMEM-
   256*INT(HMEM/256):HMEM=INT(HMEM/256)
AX 560 POKE 88,LMEM:POKE 89,HMEM:RETURN
VH 600 DL=PEEK(560)+256*PEEK(561):LMEM=PE
   EK(88):HMEM=PEEK(89)
YB 610 MEM=HMEM*256+LMEM:RETURN
```

continued on next page

```
SF 625 POKE 752,1:POSITION 7,23:? "PRESS SPACE BAR TO CONTINUE";
XH 630 GET #1,A:IF A<>32 THEN 630
ZX 635 RETURN
RT 900 POKE DL+24,130
XQ 930 POKE 512,235:POKE 513,6:POKE 54286,192
IB 940 ? " SEG.  MODE  # LINES  SCAN LINES LEFT"
ZV 950 ? " ____  ____  ________  ______________________"
XK 960 ? "  --    --       --              192"
VK 970 POKE 703,4:RETURN
QW 999 REM Create a New Display
BU 1000 GRAPHICS 0:FOR I=1 TO 16:MODE(I)=0:LINES(I)=0:NEXT I
JX 1005 POKE 752,1:LEFT=192:NSEG=0:POKE 1536,0:F=0
UD 1010 POKE 1777,28:GOSUB 900
GX 1020 CH=NSEG+1:IF CH=17 THEN GOSUB SPACE:GOTO 100
VQ 1030 GOSUB 1040:NSEG=NSEG+1:GOTO 1020
MU 1040 ? :? "Enter graphics mode for segment ";CH;
OF 1045 TRAP 1070:INPUT GRA:IF GRA=0 THEN 100
BK 1050 IF GRA>1 AND GRA<16 THEN 1080
RK 1055 GOTO 1070
VN 1060 POP :TRAP OFF:IF NSEG>0 THEN 100
OE 1065 GOTO 100
QG 1070 GOSUB NO:? "MUST BE 2-15":GOTO 1040
DF 1080 ? "How many mode lines ";:TRAP 1100:INPUT NUM
AR 1085 IF NUM=0 THEN NSEG=NSEG-1:RETURN
BM 1090 IF NUM>0 AND NUM<=(LEFT/SLPER(GRA)) THEN TRAP OFF:GOTO 1110
DE 1100 GOSUB NO:? "MUST BE 0-";INT(LEFT/SLPER(GRA)):GOTO 1080
KW 1110 MODE(CH)=GRA:LINES(CH)=NUM:LEFT=LEFT-NUM*SLPER(GRA)
PG 1120 POSITION 4,2+CH:? #6;CH:POSITION 10,2+CH:? #6;GRA
YI 1125 POSITION 18,2+CH:? #6;NUM
SI 1130 POSITION 28,2+CH:? #6;LEFT
AN 1140 RETURN
YS 1999 REM Change Current Display
YY 2000 GRAPHICS 0:IF NSEG=0 THEN GOSUB NO:POSITION 8,5:? "NO DISPLAY IN MEMORY NOW":GOSUB SPACE:GOTO 100
CF 2005 POKE 1777,200:GRAPHICS 0:POKE 752,1:GOSUB 900:F=0
ZR 2010 LEFT=192:FOR I=1 TO NSEG
QO 2020 POSITION 4,2+I:? #6;I:POSITION 10,2+I:? #6;MODE(I)
XM 2030 POSITION 18,2+I:? #6;LINES(I)
EG 2040 LEFT=LEFT-LINES(I)*SLPER(MODE(I)):POSITION 28,2+I:? #6;LEFT:NEXT I
CC 2050 ? :? "Enter segment number to change: ";
RY 2055 TRAP 2100:INPUT CH:IF CH=0 THEN 100
ZG 2060 IF CH>16 THEN 2080
SR 2062 IF CH<1 OR CH>NSEG+1 THEN 2100
VV 2065 LEFT=LEFT+LINES(CH)*SLPER(MODE(CH))
CB 2070 IF CH<>NSEG+1 THEN 2150
RC 2080 IF LEFT=0 OR NSEG=16 THEN GOSUB NO:? "DISPLAY IS FULL":GOTO 2050
LE 2090 IF CH=NSEG+1 THEN 2110
VH 2100 GOSUB NO:? "MUST BE FROM 1-";NSEG+1-(LEFT=0):GOTO 2050
QU 2110 GOSUB 1040:NSEG=CH-(NUM=0):GOTO 2170
WX 2150 GOSUB 1040:IF NUM<>0 THEN 2170
HQ 2155 IF NSEG=0 THEN POSITION 2,3:? #6;BL$(4):GOTO 2050
OG 2157 IF NSEG=15 THEN POSITION 2,18:? #6;BL$(4):GOTO 2050
NY 2160 FOR I=CH+1 TO NSEG+1:LINES(I-1)=LINES(I):MODE(I-1)=MODE(I):NEXT I
QL 2165 LINES(NSEG+1)=0:MODE(NSEG+1)=0
DS 2170 LEFT=192:FOR I=1 TO NSEG:LEFT=LEFT-LINES(I)*SLPER(MODE(I))
DM 2180 POSITION 4,2+I:? #6;I:POSITION 10,2+I:? #6;MODE(I);" "
IK 2183 POSITION 18,2+I:? #6;LINES(I);" "
VH 2185 POSITION 28,2+I:? #6;LEFT;"  ":NEXT I
SS 2190 ? #6;BL$(4):GOTO 2050
CG 2999 REM Load Display from Disk
MV 3000 GRAPHICS 0:POKE 710,16
KT 3010 POKE 82,2:? "    DISPLAYS STORED ON THIS DISK"
SG 3020 TRAP 3060:I=0:OPEN #4,6,0,"D:*.DSP"
HS 3030 INPUT #4,A$:I=I+1
EI 3040 POSITION 6,3+I
ZW 3050 ? A$(3,10):GOTO 3030
FG 3060 CLOSE #4:I=I-1:TRAP OFF
PJ 3070 IF I=0 THEN GOSUB NO:POSITION 7,6:? "NO DISPLAYS ON THIS DISK":GOSUB SPACE:GOTO 100
FE 3080 POSITION 6,4+I:? BL$
VK 3090 POSITION 2,18:? "LOAD what display:":? "(0 TO RETURN TO MENU)"
SD 3100 POSITION 21,18:INPUT A$:IF A$="0" THEN 100
NQ 3110 IF A$="" THEN GOSUB NO:GOTO 3100
XB 3120 FNAME$="D:":FNAME$(3)=A$
TB 3130 FNAME$(LEN(FNAME$)+1)=".DSP"
YE 3140 ? CHR$(125);"LOADING ";FNAME$;"...."
CM 3150 TRAP 3160:OPEN #4,4,0,FNAME$:NSEG=0:TRAP OFF:GOTO 3180
MG 3160 GOSUB NO:? :? "CANNOT LOAD FILE-ERROR ";PEEK(195):GOSUB SPACE
AU 3170 TRAP OFF:CLOSE #4:GOTO 3000
RO 3180 FOR I=1 TO 20:INPUT #4,A$
PN 3190 IF A$(4,7)="POKE" OR A$(5,8)="POKE" THEN NSEG=NSEG+1
UZ 3200 IF A$(7,9)="END" THEN INPUT #4,A$,A$,A$:POP :GOTO 3220
EY 3210 NEXT I
UI 3220 FOR I=1 TO 16:MODE(I)=0:LINES(I)=0:NEXT I
GL 3230 J=12:CT=-6
LF 3240 B$="":INPUT #4,A$:IF A$(7,10)="POKE" THEN 3290
OX 3250 IF A$(J,J)="," THEN POKE 1542+CT,VAL(B$):CT=CT+1:B$="":GOTO 3270
DU 3260 B$(LEN(B$)+1)=A$(J,J)
TD 3270 J=J+1:IF J>LEN(A$) THEN POKE 1542+CT,VAL(B$):CT=CT+1:J=12:GOTO 3240
RZ 3280 GOTO 3250
PD 3290 CLOSE #4
TQ 3300 OSMODE=PEEK(1539)-64
VV 3310 FOR I=2 TO 15:IF OSMODE=I THEN 3330
FD 3320 NEXT I
ET 3330 POP :MODE(1)=I:LINES(1)=1:J=6:Y=1
MR 3340 J=J+1:OSMODE=PEEK(1535+J)
UT 3350 IF OSMODE=65 THEN 2000
NF 3360 IF OSMODE>65 THEN OSMODE=OSMODE-64:J=J+2
LB 3370 IF OSMODE=MODE(Y) THEN LINES(Y)=LINES(Y)+1:GOTO 3340
TP 3380 FOR I=2 TO 15:IF OSMODE=I THEN 3400
FY 3390 NEXT I
BE 3400 POP :Y=Y+1:MODE(Y)=I:LINES(Y)=1:GOTO 3340
SH 5999 REM See the Current Display
OQ 6000 GRAPHICS 0:IF NSEG=0 THEN 2000
QY 6010 POKE 82,6:? :? :? "The screen will be black":? "briefly while your display"
```

```
IH 6020 ? "is created."
EP 6040 ? :? "Press any key when you are"
   :? "done viewing the display."
YY 6050 GOSUB SPACE:POKE 82,2
EJ 6100 DATA 8,8,7,7,6,5,0;0,0,1,4,0,2,3
KJ 6110 RESTORE 6100
TZ 6120 READ MAX:FOR I=1 TO NSEG:IF BAS(M
   ODE(I))=MAX THEN POP :GOTO 6140
FF 6130 NEXT I
UD 6140 GRAPHICS MAX:POKE 752,1:GOSUB LOC
   S:POKE 559,0
MP 6145 FOR I=0 TO 16:INC(I)=0:NEXT I
NE 6150 FOURK=INT(MEM/4096+1)*4096
GN 6160 POKE 1536,112:POKE 1537,112:POKE
   1538,112
WX 6170 POKE 1539,64+MODE(1)
PR 6180 POKE 1540,LMEM:POKE 1541,HMEM
WC 6185 CT=0
LJ 6190 FOR I=1 TO NSEG:TRAP 6280
UZ 6195 IF I=1 AND LINES(I)=1 THEN 6280
LA 6200 FOR J=1 TO LINES(I):IF I=1 AND J=
   1 THEN 6270
FP 6210 MEM=MEM+BPER(MODE(I)):IF MEM<=FOU
   RK THEN 6260
MX 6220 FOURK=INT(MEM/4096+1)*4096
QX 6230 POKE 1542+CT,INT(MEM/256):CT=CT+1
   :GOTO 6270
GD 6240 Y=MEM-256*INT(MEM/256):POKE 1542+
   CT,Y:CT=CT+1:INC(I)=Y
RD 6250 POKE 1542+CT,INT(MEM/256):CT=CT+1
   :GOTO 6270
ZL 6260 POKE 1542+CT,MODE(I):CT=CT+1
GE 6270 NEXT J
WD 6280 NEXT I:TRAP OFF
MV 6290 POKE 1542+CT,65
LD 6320 POKE 560,0:POKE 561,6:POKE 559,34
NG 6330 DATA 108,81,64,53,53,64
VY 6340 RESTORE 6330:FOR I=1 TO 5:READ Y:
   FOR J=1 TO 5:SOUND 0,Y,10,8
IT 6350 NEXT J:NEXT I:SOUND 0,0,0,0
CR 6360 GET #1,A
EQ 6370 IF F=1 THEN GRAPHICS 0:GOTO 7070
BE 6380 F=1:GOTO 100
ZB 6999 REM Save Display on Disk
OR 7000 GRAPHICS 0:IF NSEG=0 THEN 2000
GC 7010 ? "Enter a NAME for this display:
   ":? "(0 TO RETURN TO MENU)"
ZG 7020 POSITION 10,4:INPUT A$
OF 7030 IF A$="0" THEN 100
FQ 7040 FNAME$="D:":FNAME$(LEN(FNAME$)+1)
   =A$
TJ 7050 FNAME$(LEN(FNAME$)+1)=".DSP"
ND 7060 IF F=0 THEN F=1:GOTO 6000
JO 7070 ? :? "Storing ";FNAME$;" on disk.
   ..."
ZI 7080 TRAP 7500:OPEN #4,8,0,FNAME$
CY 7100 ? #4;"10 GOSUB 30000:LMEM=PEEK(88
   ):HMEM=PEEK(89)":LN=20
VI 7110 FOR I=1 TO NSEG
JB 7120 ? #4;LN;" POKE 87,";BAS(MODE(I))-
   (MODE(I)=14);
FY 7130 IF I=1 THEN ? #4;"":GOTO 7165
QC 7140 BYTE=LINES(I-1)*BPER(MODE(I-1))+I
   NC(I-1)
AY 7150 ? #4;":BYTE=";BYTE;
PE 7160 ? #4;":GOSUB 31000"
YU 7165 LN=LN+10:NEXT I
CM 7170 ? #4;"29999 END"
EA 7200 ? #4;"30000 POKE 106,PEEK(106)-1:
   DL=256*PEEK(106)"
AC 7210 ? #4;"30010 GRAPHICS ";MAX;":POKE
    559,0"
EF 7230 LN=30020
FA 7240 ? #4;"30020 FOR I=0 TO ";CT+6;":R
   EAD A:POKE DL+I,A:NEXT I"
EI 7250 FOR I=1 TO 11
RI 7260 LN=LN+10:? #4;LN;" DATA ";
KM 7270 FOR J=1 TO 19
PI 7280 A=PEEK(1535+20*(I-1)+J):? #4;A;
WI 7285 IF A=65 THEN ? #4;"":LN=LN+10:GOT
   O 7340
CE 7290 ? #4;",";
FM 7300 NEXT J
KK 7310 ? #4;PEEK(1535+20*I)
FH 7320 NEXT I
OE 7340 ? #4;LN;" POKE DL+4,PEEK(88):POKE
    DL+5,PEEK(89)":LN=LN+10
VJ 7350 ? #4;LN;" POKE 560,0:POKE 561,DL/
   256:POKE 559,34"
AZ 7370 ? #4;LN+10;" RETURN"
CK 7380 ? #4;"31000 HMEM=256*HMEM+LMEM+BY
   TE"
MF 7390 ? #4;"31010 LMEM=HMEM-256*INT(HME
   M/256)"
WZ 7400 ? #4;"31020 HMEM=INT(HMEM/256)"
AF 7410 ? #4;"31030 POKE 88,LMEM:POKE 89,
   HMEM:RETURN"
VN 7420 CLOSE #4:GOTO 100
LE 7500 GOSUB NO:? :? "CANNOT STORE-ERROR
   ■";PEEK(195)
OP 7510 CLOSE #4
GX 7520 GOSUB SPACE:GOTO 7000
```

LISTING 2

```
JQ 10 GOSUB 30000:LMEM=PEEK(88):HMEM=PEEK
   (89)
II 20 POKE 87,1
QI 30 POKE 87,0:BYTE=80:GOSUB 31000
HM 40 POKE 87,7:BYTE=160:GOSUB 31000
EU 50 POKE 87,0:BYTE=800:GOSUB 31000
IA 60 POKE 87,8:BYTE=160:GOSUB 31000
HE 70 POKE 87,5:BYTE=800:GOSUB 31000
CD 29999 END
UA 30000 POKE 106,PEEK(106)-1:DL=256*PEEK
   (106)
GG 30010 GRAPHICS 8:POKE 559,0
XX 30020 FOR I=0 TO 63:READ A:POKE DL+I,A
   :NEXT I
FH 30030 DATA 112,112,112,70,80,97,6,6,6,
   2,2,2,2,14,14,14,14,14,14,14
MS 30040 DATA 14,14,14,14,14,14,14,14,14,
   14,14,14,14,5,5,5,5,15,15,15
MA 30050 DATA 15,15,15,15,15,15,15,15,15,
   15,15,15,15,15,15,15,10,10,10
WG 30060 DATA 10,10,10,65
RH 30070 POKE DL+4,PEEK(88):POKE DL+5,PEE
   K(89)
VX 30080 POKE 560,0:POKE 561,DL/256:POKE
   559,34
EJ 30090 RETURN
AE 31000 HMEM=256*HMEM+LMEM+BYTE
JW 31010 LMEM=HMEM-256*INT(HMEM/256)
CR 31020 HMEM=INT(HMEM/256)
FD 31030 POKE 88,LMEM:POKE 89,HMEM:RETURN
```

LISTING 3

```
EY 1 REM MIXADD.LST
YG 2 REM BY KARL WIEGERS
QK 3 REM ANTIC MAGAZINE
GH 15 POKE 752,1
TE 25 POSITION 0,1:? #6;"GRAPHICS 1 (ANTI
   C 6)"
SN 35 POSITION 2,1:? #6;"This is Graphics
    Mode 0 (ANTIC 2)"
YV 45 COLOR 1:PLOT 0,0:DRAWTO 159,19
BH 46 COLOR 2:PLOT 0,19:DRAWTO 159,0
VV 47 COLOR 3:PLOT 0,9:DRAWTO 159,9
ET 55 POSITION 14,2:? #6;"ANTIC MODE 5"
WP 65 COLOR 1:PLOT 0,0:DRAWTO 319,19
KH 66 PLOT 0,19:DRAWTO 319,0
QV 67 PLOT 0,9:DRAWTO 319,9
CY 75 COLOR 1:PLOT 0,0:DRAWTO 79,5
BC 76 COLOR 2:PLOT 0,5:DRAWTO 79,0
DR 77 COLOR 3:PLOT 0,3:DRAWTO 79,3
```

MANIPULATING STRINGS

Article on page 32.

LISTING 1

```
RC 10 REM VTABLE1.BAS
VN 20 REM BY BRIAN Z. WEISS
RH 30 REM ANTIC MAGAZINE
BB 40 REM
GJ 50 REM REMARKS MAY PRECEDE THE DIM STA
   TEMENT
PV 60 DIM A$(20):DIM TEST$(40)
PI 70 REM DIM TWO STRING VARIABLES IN TAB
   LE
ID 80 A$="HELLO":A$(5,5)="E":REM FOR DEMO
   NSTRATION PURPOSES
QI 100 GOSUB 1000
OA 150 END
QN 200 REM
IC 210 REM THIS SUBROUTINE CAN BE APPENDE
   D TO ANY PROGRAM FOR A LIST OF VARIABL
   ES
QR 220 REM
WV 1000 GOSUB 2000
OK 1010 VNTP=PEEK(130)+256*PEEK(131):REM
   START ADDRESS OF VARIABLE NAME TABLE
SX 1030 VNTEND=PEEK(132)+256*PEEK(133)-1:
   REM END ADDRESS OF VARIABLE NAME TABLE
WB 1050 FOR X=VNTP TO VNTEND
TY 1060 BYTE=PEEK(X):REM CONTENTS OF LOCA
   TION X
XV 1070 IF BYTE<128 THEN ? CHR$(BYTE);:GO
   TO 1150:REM PART OF VARIABLE NAME
HH 1085 LNCOUNT=LNCOUNT+1
FB 1090 IF BYTE-128=36 THEN POSITION 30,L
   NCOUNT:? "STRING":GOTO 1150
FD 1100 IF BYTE-128=40 THEN POSITION 30,L
   NCOUNT:? "ARRAY":GOTO 1150
KR 1110 ? CHR$(BYTE-128);:POSITION 30,LNC
   OUNT:? "NUMERIC"
TN 1150 IF LNCOUNT<20 THEN 1200
SO 1160 ? :? "PRESS ANY KEY TO CONTINUE":
   POKE 764,255
YP 1170 IF PEEK(764)=255 THEN 1170
LZ 1180 POKE 764,255:GOSUB 2000
MK 1200 NEXT X:RETURN
OD 2000 GRAPHICS 0:? "VARIABLE NAME";:POS
   ITION 30,0:? "TYPE"
OL 2005 ? "-------------------------------
   --------":POSITION 2,3:LNCOUNT=2:RETURN
```

LISTING 2

```
ND 10 REM VTABLE2A.BAS
VN 20 REM BY BRIAN Z. WEISS
RH 30 REM ANTIC MAGAZINE
BB 40 REM
GJ 50 REM REMARKS MAY PRECEDE THE DIM STA
   TEMENT
PV 60 DIM A$(20):DIM TEST$(40)
PI 70 REM DIM TWO STRING VARIABLES IN TAB
   LE
ID 80 A$="HELLO":A$(5,5)="E":REM FOR DEMO
   NSTRATION PURPOSES
QI 100 GOSUB 1000
OA 150 END
WV 1000 GOSUB 2000
VS 1010 VVTP=PEEK(134)+256*PEEK(135):REM
   START ADDRESS OF VARIABLE VALUE TABLE
HM 1030 VVTEND=PEEK(136)+256*PEEK(137):RE
   M END ADDRESS OF VARIABLE VALUE TABLE
XN 1050 FOR X=VVTP TO VVTEND STEP 8
TY 1060 BYTE=PEEK(X):REM CONTENTS OF LOCA
   TION X
IC 1070 IF BYTE<>129 THEN NEXT X:RETURN
HE 1075 LNCOUNT=LNCOUNT+1
AX 1076 IF LNCOUNT<20 THEN 1080
TR 1077 ? :? "PRESS ANY KEY TO CONTINUE":
   POKE 764,255
IE 1078 IF PEEK(764)=255 THEN 1078
NW 1079 POKE 764,255:GOSUB 2000:LNCOUNT=L
   NCOUNT+1
WK 1080 FOR I=X TO X+7:POSITION (I-X)*5+2
   ,LNCOUNT:? PEEK(I);"  ";:NEXT I:? :NEX
   T X
BA 1090 RETURN
VU 2000 GRAPHICS 0:? "#1   #2   #3   #4
   #5   #6   #7   #8":LNCOUNT=1:RETURN
```

LISTING 3

```
NS 10 REM VTABLE2B.BAS
VN 20 REM BY BRIAN Z. WEISS
RH 30 REM ANTIC MAGAZINE
BB 40 REM
SP 50 REM REMARKS CAN PRECEDE THE DIM STA
   TEMENT
PV 60 DIM A$(20):DIM TEST$(40)
PI 70 REM DIM TWO STRING VARIABLES IN TAB
   LE
ID 80 A$="HELLO":A$(5,5)="E":REM FOR DEMO
   NSTRATION PURPOSES
QI 100 GOSUB 1000
OA 150 END
WV 1000 GOSUB 2000
YI 1010 VVTP=PEEK(134)+256*PEEK(135)
VE 1015 STARP=PEEK(140)+256*PEEK(141):REM
    START ADDRESS OF VARIABLE VALUE TABLE
PS 1030 VVTEND=PEEK(136)+256*PEEK(137):RE
   M ENDING ADDRESS OF VARIABLE VALUE TAB
   LE
XN 1050 FOR X=VVTP TO VVTEND STEP 8
TY 1060 BYTE=PEEK(X):REM CONTENTS OF LOCA
   TION X
IC 1070 IF BYTE<>129 THEN NEXT X:RETURN
HE 1075 LNCOUNT=LNCOUNT+1
AX 1076 IF LNCOUNT<20 THEN 1080
TR 1077 ? :? "PRESS ANY KEY TO CONTINUE":
   POKE 764,255
IE 1078 IF PEEK(764)=255 THEN 1078
NW 1079 POKE 764,255:GOSUB 2000:LNCOUNT=L
   NCOUNT+1
TY 1080 FOR I=X TO X+1:POSITION (I-X)*5+2
   ,LNCOUNT:? PEEK(I);"  ";:NEXT I
HM 1090 ADD=PEEK(X+2)+256*PEEK(X+3)+PEEK(
   140)+256*PEEK(141)
EN 1100 LENGTH=PEEK(X+4)+256*PEEK(X+5)
TP 1110 MAX=PEEK(X+6)+256*PEEK(X+7)
EA 1120 POSITION 12,LNCOUNT:? ADD:POSITIO
   N 22,LNCOUNT:? LENGTH:POSITION 30,LNCO
   UNT:? MAX
MR 1130 NEXT X:RETURN
CD 2000 GRAPHICS 0:? "#1   #2   ADDRESS
    LENGTH  DIMENSION":LNCOUNT=1:RETURN
```

LISTING 4

```
SE 10 REM VTABLE3.BAS
VN 20 REM BY BRIAN Z. WEISS
RH 30 REM ANTIC MAGAZINE
RU 40 DIM A$(1)
LZ 50 REM SIZE WILL BE CHANGED LATER
IW 60 VVTP=PEEK(134)+256*PEEK(135)
RK 70 GRAPHICS 0:LIST :REM FOR DEMONSTRAT
   ION PURPOSES
DP 100 DL=PEEK(560)+256*PEEK(561):SCRN=PE
   EK(DL+4)+256*PEEK(DL+5)
FL 150 OFFSET=SCRN-PEEK(140)-PEEK(141)*25
   6
FS 160 FOUR=INT(OFFSET/256)
SU 170 THREE=OFFSET-FOUR*256
UP 200 POKE VVTP+2,THREE:POKE VVTP+3,FOUR
WI 250 SIZE=400
JA 260 SIX=INT(SIZE/256)
DW 270 FIVE=SIZE-SIX*256
YC 280 POKE VVTP+4,FIVE:POKE VVTP+6,FIVE
RO 290 POKE VVTP+5,SIX:POKE VVTP+7,SIX
ES 400 A$=CHR$(0):A$(400)=CHR$(0):A$(2)=A
   $
CG 420 A$(41,45)="(%,,/"
QK 430 FOR DEL=1 TO 500:NEXT DEL
QZ 450 REM
LR 460 REM THIS ROUTINE SHOWS SOME USES F
   OR A STRING LOCATED IN SCREEN MEMORY
SK 500 GRAPHICS 0
ID 510 A$=CHR$(0):A$(400)=A$:A$(2)=A$
MK 520 FOR B=1 TO 5:FOR A=34 TO 58
QU 530 A$=CHR$(A):A$(400)=A$:A$(2)=A$
KD 540 FOR DEL=B*10 TO 50:NEXT DEL:NEXT A
RJ 550 FOR DEL=1 TO 200:NEXT DEL:NEXT B
JW 560 FOR B=1 TO 2
IP 570 A$=CHR$(0):A$(400)=A$:A$(2)=A$
PN 580 A$(365,399)="moving this string ar
   ound is easy"
SU 585 A$(371,371)=CHR$(0):A$(376,376)=CH
   R$(0):A$(383,383)=CHR$(0):A$(390,390)=
   CHR$(0):A$(393,393)=CHR$(0)
OV 590 FOR DEL=1 TO 200:NEXT DEL
JD 600 FOR A=364 TO 161 STEP -1:A$(A,A+34
   )=A$(A+1,A+35):FOR DEL=B*15 TO 30:NEXT
    DEL:NEXT A:NEXT B:GRAPHICS 0:END
```

now you can save and edit your Info Bits files

SON OF INFO BITS

Article on page 45.

LISTING 1

```
UN 5 REM INFONEW.BAS
IF 6 REM by ANDY BARTON
QO 7 REM ANTIC MAGAZINE
HM 8 REM DELETE ENTRY ROUTINE AND UPGRADE
GV 9 REM FOR INFO BITS
LM 10 ? :? :? "  loading  INFO BITS"
MX 15 GOSUB 2000
FY 20 POKE 82,0
RY 30 OPEN #2,12,0,"S:"
WZ 40 ? "↖":DIM B$(130),R$(120),SEARCH$(1
   20),RECORD$(120),S$(120)
VV 50 ? :? :? "  INFO BITS"
ZS 60 TRAP 60:? :? "  1) ADD TO FILE   2)
   SEARCH FOR ENTRY"
RH 61 ? "                3) DELETE/(EDIT) ENTR
   Y"
RW 70 INPUT X
HQ 80 ON X GOTO 110,200,300
LF 100 REM ADD TO FILES
SE 110 OPEN #3,9,0,"D1:INFOBITS.FIL"
YS 120 ? :? "TYPE ENTRY:"
CZ 125 INPUT #5;B$
LC 130 IF B$="" THEN CLOSE #3:GOTO 60
KF 140 ? #3;B$:GOTO 120
LG 200 REM SEARCH
PJ 205 OPEN #1,4,0,"D1:INFOBITS.FIL"
ES 210 ? :? "SEARCH FOR:":INPUT #5;SEARCH
   $
MF 220 IF SEARCH$="ALL" THEN 1000
SL 230 GOSUB 500
NY 240 CLOSE #1:GOTO 60
YC 300 REM DELETE ENTRY
RC 310 ? "↖"
XP 320 ? :? "PRESS  S  SEARCH FOR AN ENTR
   Y                 D  DELETE/(EDIT) LA
   ST ENTRY ABOVE        A  ABORT"
AA 330 INPUT B$
RR 340 IF B$="A" THEN 60
RE 350 IF B$="S" THEN 450
YG 360 IF B$<>"D" THEN ? "↑↑":GOTO 330
PJ 370 OPEN #1,4,0,"D1:INFOBITS.FIL"
SS 380 OPEN #4,8,0,"D1:INFOBITS.FIL"
BN 385 ? "↖":? :? "  I'm working on it":?
    :?
TH 390 POKE 752,1:POKE 206,0:POKE 207,0:X
   =USR(1737):POKE 752,0
PC 400 ? :? "   ABOVE ENTRY DELETED "
ZD 430 CLOSE #1:CLOSE #4
NT 435 ? :? "EDIT/RE-ENTER DELETED ENTRY
   ABOVE? (Y)"
EE 440 INPUT B$:IF B$<>"Y" THEN 60
VH 445 ? :? "↑↑↑↑↑↑↑↑↑":OPEN #3,9,0,"D1:I
   NFOBITS.FIL":POSITION 0,4:? "EDIT ENTR
   Y:":GOTO 125
PG 450 OPEN #1,4,0,"D1:INFOBITS.FIL"
FE 460 ? :? "SEARCH FOR:":INPUT #5;SEARCH
   $
SV 470 GOSUB 500
AL 480 CLOSE #1:GOTO 320
ZC 500 ? :POKE 752,1:POKE 206,0:POKE 207,
   0:POKE 226,LEN(SEARCH$):X=USR(1536,ADR
   (SEARCH$)):POKE 752,0:RETURN
AX 1000 REM ^^^^^ PRINT ALL ENTRIES
NK 1010 TRAP 240
PJ 1020 INPUT #1;RECORD$:? RECORD$:GOTO 1
   020
QI 1030 INPUT #1,B$:RECORD$=B$(11,LEN(B$)
   )
UK 1040 ? RECORD$:NEXT X
EB 2000 FOR A=1536 TO 1791:READ B:POKE A,
   B:NEXT A
KL 2001 DATA 104,104,133,225,104,133,224,
   162,16,32,168,6,162,16,32,184,6,32,86,
   228,48,92,162,0,160
```

continued on next page

```
WD 2002 DATA 0,189,253,3,42,16,2,41,191,1
   06,209,224,240,11,192,0,240,12,160,0,1
   66,227,76,55,6
CR 2003 DATA 200,196,226,176,15,232,236,8
   8,3,176,207,192,0,208,217,134,227,76,2
   6,6,32,76,6,76,12
XF 2004 DATA 6,162,32,169,9,157,66,3,32,1
   90,6,165,84,201,20,176,23,169,253,157,
   68,3,169,3,157
RY 2005 DATA 69,3,32,86,228,165,206,133,2
   08,165,207,133,209,96,96,169,156,157,6
   8,3,169,6,157,69,3
EB 2006 DATA 32,86,228,173,252,2,201,255,
   240,249,169,255,141,252,2,169,166,157,
   68,3,169,6,157,69,3
WY 2007 DATA 32,86,228,76,76,6,160,193,20
   6,217,160,203,197,217,160,155,125,155,
   169,5,157,66,3,169,253
FL 2008 DATA 157,68,3,169,3,157,69,3,96,2
   30,206,208,2,230,207,169,122,157,72,3,
   169,0,157,73,3
NP 2009 DATA 96,104,162,64,169,9,157,66,3
   ,32,173,6,162,16,32,168,6,162,16,32,18
   4,6,32,86,228
JZ 2010 DATA 48,143,165,206,197,208,208,1
   2,165,207,197,209,208,6,32,76,6,76,217
   ,6,162,64,32,190,6
RC 2011 DATA 32,86,228,76,217,6
CR 2012 OPEN #5,4,0,"E:":RETURN
```

LISTING 2

```
ID 10 REM INFOMOD.BAS
PY 20 REM BY ANDY BARTON
RH 30 REM ANTIC MAGAZINE
SX 40 DIM IN$(131),OUT$(122)
SD 50 OPEN #1,4,0,"D1:INFOBITS.FIL"
UK 60 OPEN #2,8,0,"D1:INFOBITS.FIL"
KI 70 TRAP 100
NW 80 INPUT #1;IN$:OUT$=IN$(11,LEN(IN$))
ZM 90 ? #2;OUT$:GOTO 80
KR 100 CLOSE #1:CLOSE #2:END
```

Extra-convenient menu program

LAZY LOADER

Article on page 35.

LISTING 1

```
UX 1 REM LAZY LOADER
PY 2 REM BY FRANK WALTERS
QK 3 REM ANTIC MAGAZINE
RH 10 DIM D$(13),DIR$(6),F$(15),T$(200),M
   $(136),Z$(64):D=49:DIR$="D1:*.*":F$="D
   1:"
BD 15 Z$="2→←┴┴↓£350B ┣6▗ 7fu?▃▔:*8=9▔┣┓♥%#
   ◢/()-"+.|+┬♦FbGN⌧<▃▔:*8=9▔┣┓♥%#◢"
LI 20 FOR I=0 TO 128 STEP 8:FOR J=1 TO 7:
   M$(I+J)=CHR$(254):NEXT J:M$(I+J,I+J)=C
   HR$(29):NEXT I
KV 30 POKE 82,0:POKE 83,39:GOSUB 40:GOTO
   300
CF 40 T$(1)="┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━
   ━━━━━━━━━━━━━━━━━┓| LAZY LOADER       by Fr
   ank Walters     ┃"
IB 50 T$(81)="┃ <1-4>Drive#:          <RETURN
   >Directory  ||  Keys:           RUN:
   ████████████████████████┃"
KH 60 T$(161)="┗━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━
   ━━━━━━━━━━━━━━━━━━━━━━━━┛":RETURN
QC 70 T$(129,132)="CAPS":IF PEEK(702)=0 T
   HEN T$(129,132)="lowr"
OT 80 IF PEEK(702)=128 THEN T$(129,132)="
   CTRL"
TW 90 IF PEEK(694) THEN T$(129,132)="INVS
   "
ZK 100 T$(95,95)=CHR$(D+128):DIR$(2,2)=CH
   R$(D):POSITION 0,0:? T$:RETURN
WK 110 TRAP 670:CLOSE #2:OPEN #2,6,0,DIR$
VA 120 POKE 82,14:POSITION 14,5:FOR I=65
   TO 90
CE 130 INPUT #2;D$
LZ 140 IF D$(11,13)="SYS" OR D$(11,13)="E
   XE" OR D$(11,13)="OBJ" OR D$(11,13)="D
   AT" THEN 130
OH 150 IF D$(4,7)=" FRE" OR D$(5,8)=" FRE
   " THEN POP :GOTO 270
SB 160 IF I=82 THEN POKE 82,0:POSITION 0,
   5:? M$:POKE 82,20:POSITION 20,5
JD 170 ? CHR$(I+128);CHR$(190);D$(3):NEXT
    I
SF 180 FOR I=97 TO 122
CQ 190 INPUT #2;D$
FU 200 IF D$(11,13)="SYS" OR D$(11,13)="E
   XE" OR D$(11,13)="OBJ" OR D$(11,13)="D
   AT" THEN 190
OA 210 IF D$(4,7)=" FRE" OR D$(5,8)=" FRE
   " THEN POP :GOTO 270
XF 220 IF I=105 THEN POKE 82,0:POSITION 0
   ,5:? M$:POKE 82,26:POSITION 26,5
CG 230 IF I=122 THEN 250
AG 240 ? CHR$(I+128);CHR$(190);D$(3)
QA 250 NEXT I:I=I-1:IF D$(4,7)=" FRE" OR
   D$(5,8)=" FRE" THEN 270
SP 260 POSITION 12,22:? "more files...":C
   LOSE #2:TRAP 40000:POKE 82,0:I=I-1:RET
   URN
XC 270 CLOSE #2:TRAP 40000:POKE 82,0:I=I-
   1
SB 280 POSITION 12,22:? D$;:IF D$(5,5)="
   " THEN ? "T";
ZD 290 ? "ORS";:RETURN :REM CHANGE 'ORS'
   TO 'KS' FOR DOS 3
LT 300 GRAPHICS 0:POKE 752,1:POKE 710,192
   :POKE 709,198:POKE 712,192
LQ 301 GOSUB 70:GOSUB 110
UA 302 POKE 694,0:IF PEEK(702)=128 THEN P
   OKE 702,64
VL 303 GOSUB 70
YC 304 POKE 764,255:FLG=1
FQ 305 IF PEEK(764)=255 THEN 305
WC 306 IF PEEK(764)=124 OR PEEK(764)=60 T
   HEN POKE 702,(PEEK(764)-60):GOSUB 70:G
   OTO 304
NU 307 N=PEEK(764):IF N>63 THEN N=N-64:FL
   G=0
DU 308 IF PEEK(702)=64 THEN FLG=0
MV 309 IF PEEK(764)=12 THEN RUN
UG 310 FOR X=1 TO 64:IF CHR$(N)<>Z$(X,X)
```

```
    THEN 314
XI 311 IF X<11 THEN FLG=0
HB 312 K=(X+47)+32*(FLG):POP :GOTO 350
LF 314 NEXT X:GOTO 303
FX 350 POKE 764,255:GOSUB 70:IF K>48 AND
   K<53 THEN D=K:GOTO 300
UO 360 IF K=4 THEN POKE 82,2:GRAPHICS 0:D
   OS
PX 390 IF K<65 THEN 350
XJ 400 IF K>90 AND K<97 THEN 350
GY 410 IF K>I THEN 350
CU 420 POSITION 19,3:? CHR$(K+128)
FB 430 CLOSE #1:T$(140,140)=CHR$(K+128):G
   OSUB 70
AP 440 F=K-64:IF F>26 THEN F=F-6
IG 450 CLOSE #2:OPEN #2,6,0,DIR$
XF 460 TRAP 670:FOR I=1 TO F
CP 470 INPUT #2;D$
JS 480 IF D$(11,13)="SYS" OR D$(11,13)="E
   XE" OR D$(11,13)="OBJ" OR D$(11,13)="D
   AT" THEN 470
AJ 490 NEXT I:CLOSE #2:D$=D$(3,LEN(D$)):F
   $(2,2)=CHR$(D)
JQ 500 IF D$(9,9)<>" " THEN 530
KS 510 FOR I=1 TO 9:IF D$(I,I)=" " THEN P
   OP :GOTO 570
BE 520 F$(I+3)=D$(I,I):NEXT I
PT 530 FOR I=1 TO 8
FS 540 IF D$(I,I)=" " THEN POP :GOTO 560
BK 550 F$(I+3)=D$(I,I):NEXT I
JN 560 F$(I+3,I+3)=".":FOR J=9 TO 11:IF D
   $(J,J)<>" " THEN F$(I+4)=D$(J,J):I=I+1
   :NEXT J
NG 570 T$(142,156)="               ":T$(1
   42,141+LEN(F$))=F$:GOSUB 70
BQ 580 TRAP 590:POKE 82,2:POKE 702,64:GRA
   PHICS 0:RUN F$
MM 590 IF PEEK(195)<>21 THEN 670
AM 600 POKE 82,0:GOSUB 710
XC 610 CLOSE #1:OPEN #1,4,0,"K:"
GF 620 GET #1,K:IF K<>89 AND K<>78 THEN 6
   20
WF 630 CLOSE #1:IF K=78 THEN 30
JX 640 ? CHR$(125):GOSUB 70:POKE 82,2:POK
   E 752,0:POSITION 15,3:? "ENTER":POSITI
   ON 2,10
ZB 650 TRAP 40000:? "PRESS  RETURN  WHEN"
ZP 660 ? "        TO":? " ENTER ";CHR$(34);
   F$:? :? :? " RUN";"↑↑↑↑↑";:NEW
JJ 670 TRAP 40000
FO 680 POSITION 20,2:? "<RETURN>":POSITIO
   N 21,3:? "  ERROR - ";PEEK(195);"   "
PN 690 IF PEEK(764)<>12 THEN 690
MV 700 POKE 195,0:POKE 764,255:RUN
QF 710 COLOR 160:X=11:Z=27:FOR Y=8 TO 16:
   PLOT X,Y:DRAWTO Z,Y:NEXT Y:PLOT Z,Y-1
HG 720 POSITION 14,9:? "DO YOU WANT":POSI
   TION 15,11:? "TO ENTER:"
AW 730 POSITION 19-(LEN(F$)/2),13:? F$
NV 740 POSITION 17,15:? "(Y/N) ":RETURN
```

automatically run the program of your choice

AUTORUN.SYS

Article on page 35.

LISTING 1

```
JN 10 REM ARSMAKER.BAS
LJ 20 REM - ABC'S OF THE ATARI COMPUTERS
QA 30 REM REPRINTED IN ANTIC MAGAZINE
OL 40 GRAPHICS 0:DIM A$(128),B$(12)
HP 50 ? :? "  This program creates a"
SD 60 ? "disk file called AUTORUN.SYS"
SO 70 ? "which will RUN a SAVEd BASIC pro
   gram.":? "when the disk is booted.":?
QD 80 ? "  To create an AUTORUN.SYS file
   for"
ZE 90 ? "the program ";CHR$(34);"D:MENU";
   CHR$(34);", for example,"
RJ 100 ? "you'd type  MENU  and press [RE
   TURN].":?
HR 110 ? "  The resulting AUTORUN.SYS fil
   e":? "would RUN any program called MEN
   U.":?
EE 120 ? :? "ENTER FILENAME TO AUTORUN";:
   INPUT B$
YE 130 A$(1,6)="RUN D:":A$(4,4)=CHR$(34):
   A$(7,7+LEN(B$))=B$:A$(7+LEN(B$))=CHR$(
   34)
WC 140 OPEN #1,8,0,"D:AUTORUN.SYS"
OR 150 ? #1;"▮▮♥/";
VY 160 L=123+LEN(A$)-1
MS 170 PUT #1,L
CQ 180 PUT #1,6
BT 190 FOR I=1 TO 123
TL 200 READ D
KM 210 IF I=64 THEN PUT #1,LEN(A$)-1:GOTO
    230
IR 220 PUT #1,D
FY 230 NEXT I
QI 240 FOR I=LEN(A$) TO 1 STEP -1
AN 250 PUT #1,ASC(A$(I,I))
GE 260 NEXT I
QJ 270 ? #1;"▮▮▮ ▮ |♥/";
LO 280 CLOSE #1
OJ 290 END
EE 300 DATA 162,0,189,26,3,201,69,240,5,2
   32
IS 310 DATA 232,232,208,244,232,142,105,6
   ,189,26
EE 320 DATA 3,133,205,169,107,157,26,3,23
   2,189
LM 330 DATA 26,3,133,206,169,6,157,26,3,1
   60
WO 340 DATA 0,162,16,177,205,153,107,6,20
   0,202
XI 350 DATA 208,247,169,67,141,111,6,169,
   6,141
MN 360 DATA 112,6,169,10,141,106,6,96,172
   ,106
EZ 370 DATA 6,240,9,185,123,6,206,106,6,1
   60
YH 380 DATA 1,96,138,72,174,105,6,165,205
   ,157
FD 390 DATA 26,3,232,165,206,157,26,3,104
   ,170
HB 400 DATA 169,155,160,1,96,0,0,0,0,0
ZY 410 DATA 0,0,0,0,0,0,0,0,0,76
XE 420 DATA 0,0,0
```

enhanced dot-by-dot picture dissolves

FADER II

Article on page 57.

LISTING 1

```
ZS 10 REM FADERII.BAS
QB 20 REM BY PATRICK DELL'ERA
RH 30 REM ANTIC MAGAZINE
ZV 40 CLR :DIM FN$(20),TEMP$(20),AR$(93)
KI 50 CLOSE #1:GRAPHICS 0:? :? :? :TRAP 4
   0000:POKE 710,98
VM 60 ? "1. Insert a DOS 2.0 disk":? "
   into your drive."
JJ 70 ? :? "2. Type in the filename of th
   e":? "    resulting object file.":? :IN
   PUT FN$
MH 80 TRAP 150
FA 90 IF LEN(FN$)<3 THEN 110
QK 100 IF FN$(1,1)="D" AND (FN$(2,2)=":"
   OR FN$(3,3)=":") THEN 140
BP 110 TEMP$(1,2)="D:":TEMP$(3)=FN$:FN$=T
   EMP$
PS 120 TRAP 130:OPEN #1,4,0,"D:DOS.SYS":C
   LOSE #1:GOTO 140
CC 130 ? :? " INSERT A DOS 2.0 DISK ! ":F
   OR X=1 TO 250:NEXT X:GOTO 50
UK 140 TRAP 150:OPEN #1,8,0,FN$:CLOSE #1:
   GOTO 170
GJ 150 ? :? " ILLEGAL FILENAME ! ":FOR X=
   1 TO 250:NEXT X:GOTO 50
KJ 170 ? :? " ... Creating file, please w
   ait ":FLAG=0:TRAP 182
UG 180 RESTORE :READ LN:C=1:DIM A$(LN)
ET 181 TRAP 230:GOTO 190
IR 182 FLAG=1:POKE 712,4:OPEN #1,8,0,FN$
BW 190 AR$="":READ AR$
JX 200 FOR X=1 TO LEN(AR$) STEP 3
TO 215 IF FLAG THEN PUT #1,VAL(AR$(X,X+2)
   ):NEXT X:TRAP 280:GOTO 190
HD 220 A$(C,C)=CHR$(VAL(AR$(X,X+2))):C=C+
   1:NEXT X:GOTO 190
KH 230 NUMHI=INT(LN/256):NUMLO=LN-NUMHI*2
   56
AT 240 OPEN #1,8,0,FN$
UA 250 AD=ADR(A$):ADHI=INT(AD/256):ADLO=A
   D-ADHI*256
RK 260 IO=848:POKE IO+2,11:POKE IO+4,ADLO
   :POKE IO+5,ADHI:POKE IO+8,NUMLO:POKE I
   O+9,NUMHI
TK 270 X=USR(ADR("hhh LV "),16)
TV 280 CLOSE #1:POKE 712,0:? " COMPLETED
    "
OM 1000 DATA 988
DF 1002 DATA 2552550000622510620020492302
   24076008063169255133224032002062166224
   224026240038224013240009224
WY 1004 DATA 0072082391332340760110621332
   05032002062133206032002062133207032002
   062133208032002062133204076
LH 1006 DATA 0110621690001332361332301650
   88133224133228165089133225133229032008
   063072162000134227041128133
XZ 1008 DATA 2351040411271332262080100320
   08063133227032008063133226198226165235
   208042032008063133233076158
TK 1010 DATA 0621982261692551972262082451
   98227169255197227208237240195198226169
   255197226208008198227169255
XF 1012 DATA 1972272401790320080631332331
   69002197234240082240209165233160000145
   224024169080101224133224169
UO 1014 DATA 0001012251332252302301690961
   97230208047169001197236208024024169001
   101228133228133224169000133
MQ 1016 DATA 2361332301012291332291332250
   76240062230236024169040101228133224169
   000133230101229133225165235
HZ 1018 DATA 2401762081471652331600001452
   24252062247063230224208002230225165235
   240158208236162016169000157
CN 1020 DATA 0720031570730030320862280480
   01096104104076135065162016169012157066
   003076086228157069003152157
WF 1022 DATA 0680031690031570660030320862
   28048001096032029063108010000001008064
   002016128004032165088133128
GP 1024 DATA 1331301690001331871690071971
   87176001096166187169000141040032169254
   205040032144019138072174040
BA 1026 DATA 0321880430321041700321840632
   38040032076095063138072162255188043032
   104170032184063173011032048
EP 1028 DATA 0411641871920022080061852030
   00141196002192003208006185203000141197
   002192004208006185203000141
AN 1030 DATA 1980021920052080051652041412
   00002230187076081063173009032133129173
   010032133131169000141041032
NW 1032 DATA 1690292050410321760010961771
   28061063063208010189063063073255049130
   076226063017130145130230129
NP 1034 DATA 2301312321380410071702380410
   32076199063173001062141185248063243064
   065162001134009134203202142
LB 1036 DATA 0680021340651381570430322322
   08249142040032169255133133077040032072
   133132010176005070133076025
VZ 1038 DATA 0641730102100371331971322400
   02176245141042032170189043032168104072
   170189043032174042032157043
VB 1040 DATA 0321041701521570430322380400
   32230203208194162096032031063169012157
   074003169008157075003160193
BE 1042 DATA 1690650320390631691281412000
   02173048002133128173049002133129160003
   169078145128160006169014145
TK 1044 DATA 1282001920992082471690781451
   28200200200177128201015208007169014145
   128200208243141010032165089
YM 1046 DATA 1410100321730100320562330321
   41009032173009032133089169001133186032
   029063169006157074003169000
GW 1048 DATA 1570750031601841690650320390
   63132187165186197187144036162016169019
   157068003169032157069003169
CL 1050 DATA 0201570720031690001570730031
   69005157066003032086228048192230187208
   214032029063244064195065173
BN 1052 DATA 0200322010322081781620021891
   84065157018032202016247232224008240007
   189021032201032208244172029
PK 1054 DATA 0321690461570210321522321570
```

```
   21032133203169073232157021032169067232
   157021032169155232157021032
NG 1056 DATA 032029063169004157074003169
   0001570750031600181690320320390631730
   09032133089169007162016157066
JM 1058 DATA 003164203192080208003076007
   0621650881570680031650891570690031690
   00157072003169030157073003032
AE 1060 DATA 051063032008063133204032008
   0631332050320080631332060320080631332
   07032071063032029063169000133
CD 1062 DATA 077133019165019205000062176
   0251730312082010052080030760570632010
   03208007169006205031208208251
EJ 1312 DATA 201006208224230186076177064
   0680490580420460630730671550830581552
   24002225002243063
```

LISTING 2

```
UQ 5 REM FADER II MODIFIER
QA 10 REM BY PATRICK DELL'ERA
RP 15 REM ANTIC MAGAZINE
MB 20 DIM FILE$(12),FILENAME$(16)
OE 30 CLOSE #1:OPEN #1,4,0,"K:"
RU 40 GRAPHICS 0:POKE 752,1:POKE 712,144:
   POKE 710,146:POKE 708,150
IX 50 POSITION 10,1:? ;" FADER II MODIFIE
   R "
QW 60 POSITION 5,4:POKE 82,5
CE 70 ? ;" ENTER NAME OF FADER II FILE:"
CW 80 X=9:Y=6
WD 90 FILENAME$="D1:------------":POSITIO
   N X-3,Y:? FILENAME$
YB 100 FILE$=""
BT 110 GET #1,A
IO 120 IF LEN(FILE$)=12 AND A<>ASC("◄") T
   HEN ? "◙":GOTO 110
DJ 130 OKAY=0:IF (A>64 AND A<91) OR A=46
   OR (A>47 AND A<58) THEN OKAY=1
LI 140 IF OKAY THEN POSITION X,Y:? CHR$(A
   );:FILE$(LEN(FILE$)+1)=CHR$(A):X=X+1:G
   OTO 110
YS 150 IF A=155 THEN 200
FP 160 IF A<>ASC("◄") THEN 110
OJ 170 IF LEN(FILE$)=1 THEN X=9:FILE$="":
   POSITION X,Y:? #6;"-";:GOTO 110
WL 180 IF  NOT LEN(FILE$) THEN 110
AP 190 X=X-1:POSITION X,Y:? #6;"-";:POSIT
   ION X,Y:FILE$=FILE$(1,LEN(FILE$)-1):GO
   TO 110
XC 200 IF FILE$(LEN(FILE$))="-" THEN FILE
   $=FILE$(LEN(FILE$)-1):GOTO 200
NS 210 FILENAME$(4)=FILE$
HY 220 TRAP 470:CLOSE #2:OPEN #2,12,0,FIL
   ENAME$
EU 230 NOTE #2,SECTOR,BYTE:BYTE=6:POINT #
   2,SECTOR,BYTE
CV 240 GET #2,WAIT
EV 250 GET #2,DRVNUM
JT 260 POSITION 5,9:? " DRIVE # = ";CHR$(
   DRVNUM)
XM 270 POSITION 5,11:? " PAUSE TIME = ";I
   NT(WAIT*4.27+0.05);" seconds◄◄◄"
RD 280 POSITION 5,13
EE 290 ? " OPTION CHANGES DRIVE #         "
WS 300 ? "                                "
WD 310 ? " SELECT CHANGES PAUSE TIME "
XK 320 ? " (press '-' for less time, "
ZA 330 ? " any other key for more     "
SO 340 ? " pause time)                "
XC 350 ? "                                "
NA 360 ? " START SAVES CHANGES         "
OH 370 A=PEEK(53279):IF A=7 THEN 370
NF 380 A=A-2:ON A GOTO 390,370,410,440,37
   0
AE 390 DRVNUM=DRVNUM+1-(DRVNUM=52)*4
OH 400 GOTO 260
SG 410 IF PEEK(764)=14 THEN WAIT=WAIT-1+(
   WAIT=0)*256:GOTO 260
DY 420 WAIT=WAIT+1-(WAIT=255)*256
ON 430 GOTO 260
TY 440 POINT #2,SECTOR,BYTE:PUT #2,WAIT:P
   UT #2,DRVNUM
ST 450 GRAPHICS 0
KZ 460 CLOSE #2:CLOSE #1:POKE 82,2:POKE 7
   52,0:END
FA 470 IF PEEK(195)=170 THEN POSITION 6,1
   5:? "◙ Can't find that file...":GOTO 8
   0
SZ 480 GRAPHICS 0
DB 490 ? "◙↓↓Error #";PEEK(195);" at line
    #";PEEK(186)+256*PEEK(187);".":GOTO 4
   60
```

LISTING 3

```
0100 ;FADERII.M65
0101 ;BY PATRICK DELL'ERA
0102 ;ANTIC MAGAZINE
0104 ;
0105 PROGRAM = $3E00
0106 BUFFER = $2007
0107 ;External reference equates
0108 LBL:105 = BUFFER+2
0109 LBL:106 = BUFFER+3
0110 LBL:091 = BUFFER+4
0111 LBL:064 = BUFFER+$0B
0112 LBL:121 = BUFFER+$0D
0113 LBL:125 = BUFFER+$0E
0114 LBL:127 = BUFFER+$16
0115 LBL:080 = BUFFER+$21
0116 LBL:107 = BUFFER+$22
0117 LBL:083 = BUFFER+$23
0118 LBL:078 = BUFFER+$24
0119 ;End of external references
0120 ;
0121 ;System equates used
0122 BOOT =    $09
0123 DOSVEC = $0A
0124 RTCLOK = $12
0125 SOUNDR = $41
0126 ATRACT = $4D
0127 SAVMSC = $58
0128 SDLSTL = $0230
0129 SDLSTH = $0231
0130 COLDST = $0244
0131 RUNADR = $02E0
0132 COLOR0 = $02C4
0133 COLOR1 = $02C5
0134 COLOR2 = $02C6
0135 COLOR4 = $02C8
0136 ICCOM = $0342
0137 ICBAL = $0344
0138 ICBAH = $0345
0139 ICBLL = $0348
0140 ICBLH = $0349
0141 ICAX1 = $034A
0142 ICAX2 = $034B
0143 CONSOL = $D01F
0144 RANDOM = $D20A
0145 CIOV =   $E456
0146 EOL =    $9B
0147 ;End of system equates
0148 ;Zero-page equates
0149 PIC.CTR = $E0
0150 N09 =    $EA
```

continued on next page

```
0151 CTR1 =   $BA
0152 CTR2 =   $BB
0153 SVCOLR = $CC
0154 PICTYPE = $CB
0155 N16 =    $EC
0156 N17 =    $E6
0157 N18 =    $E4
0158 N19 =    $E1
0159 N22 =    $E3
0160 N23 =    $EB
0161 N24 =    $E2
0162 N27 =    $E9
0163 N41 =    $80
0164 N42 =    $81
0165 N54 =    $82
0166 N55 =    $83
0167 N59 =    $84
0168 N60 =    $85
0169 ;End of zero-page equates
0170      .PAGE "Main Program  11/10/84
"
0171 ;
0172 ;Program Start
0173 ; First two bytes are variables
0174 ; indicating the Pause length
0175 ; and the Drive # pictures are
0176 ; read from.
0177 ;
0178      *=  PROGRAM
0179 WAIT
0180      .BYTE 2
0181 DRNUM
0182      .BYTE '1
0183 ;
0184 ;Routine to load compressed
0185 ; files.
0186 ;
0187 READ.HDR
0188      INC PIC.CTR
0189      JMP GET.BYTE
0190 PIC.LDR
0191      LDA #$FF
0192      STA PIC.CTR
0193 LBL:007
0194      JSR READ.HDR
0195      LDX PIC.CTR
0196      CPX #$1A
0197      BEQ LBL:006
0198      CPX #$0D
0199      BEQ SVCOLR.PIC
0200      CPX #7
0201      BNE LBL:007
0202      STA N09
0203      JMP LBL:007
0204 SVCOLR.PIC
0205      STA SVCOLR+1
0206      JSR READ.HDR
0207      STA SVCOLR+2
0208      JSR READ.HDR
0209      STA SVCOLR+3
0210      JSR READ.HDR
0211      STA SVCOLR+4
0212      JSR READ.HDR
0213      STA SVCOLR
0214      JMP LBL:007
0215 LBL:006 LDA #0
0216      STA N16
0217      STA N17
0218      LDA SAVMSC
0219      STA PIC.CTR
0220      STA N18
0221      LDA SAVMSC+1
0222      STA PIC.CTR+1
0223      STA N18+1
0224 LBL:030 JSR GET.BYTE
0225      PHA
0226      LDX #0
0227      STX N22
0228      AND #$80
0229      STA N23
0230      PLA
0231      AND #$7F
0232      STA N24
0233      BNE LBL:025
0234      JSR GET.BYTE
0235      STA N22
0236      JSR GET.BYTE
0237      STA N24
0238 LBL:025 DEC N24
0239      LDA N23
0240      BNE LBL:026
0241      JSR GET.BYTE
0242      STA N27
0243 LBL:029
0244      JMP LBL:028
0245 LBL:032 DEC N24
0246      LDA #$FF
0247      CMP N24
0248      BNE LBL:029
0249      DEC N22
0250      LDA #$FF
0251      CMP N22
0252      BNE LBL:029
0253      BEQ LBL:030
0254 LBL:036 DEC N24
0255      LDA #$FF
0256      CMP N24
0257      BNE LBL:026
0258      DEC N22
0259      LDA #$FF
0260      CMP N22
0261      BEQ LBL:030
0262 LBL:026 JSR GET.BYTE
0263      STA N27
0264 LBL:028 LDA #2
0265      CMP N09
0266      BEQ LBL:031
0267 LBL:035 BEQ LBL:032
0268      LDA N27
0269      LDY #0
0270      STA (PIC.CTR),Y
0271      CLC
0272      LDA #$50
0273      ADC PIC.CTR
0274      STA PIC.CTR
0275      LDA #0
0276      ADC PIC.CTR+1
0277      STA PIC.CTR+1
0278      INC N17
0279      LDA #$60
0280      CMP N17
0281      BNE LBL:033
0282      LDA #1
0283      CMP N16
0284      BNE LBL:034
0285      CLC
0286      LDA #1
0287      ADC N18
0288      STA N18
0289      STA PIC.CTR
0290      LDA #0
0291      STA N16
0292      STA N17
0293      ADC N18+1
0294      STA N18+1
0295      STA PIC.CTR+1
0296      JMP LBL:033
0297 LBL:034 INC N16
0298      CLC
0299      LDA #$28
0300      ADC N18
0301      STA PIC.CTR
0302      LDA #0
0303      STA N17
```

```
0304      ADC N18+1
0305      STA PIC.CTR+1
0306 LBL:033 LDA N23
0307      BEQ LBL:035
0308 LBL:037 BNE LBL:036
0309 LBL:031 LDA N27
0310      LDY #0
0311      STA (PIC.CTR),Y
0312      INC PIC.CTR
0313      BNE LBL:031.1
0314      INC PIC.CTR+1
0315 LBL:031.1
0316      LDA N23
0317      BEQ LBL:035 ;Forced branch
0318      BNE LBL:037 ;
0319 ;
0320 ;CIO utilities
0321 ;
0322 GET.BYTE
0323      LDX #$10
0324      LDA #0
0325      STA ICBLL,X
0326      STA ICBLH,X
0327      JSR CIOV
0328      BMI LBL:038
0329      RTS
0330 LBL:038 PLA
0331      PLA
0332      JMP MIC.1
0333 CLOSE.CH1
0334      LDX #$10
0335 CLOSE.CH6
0336      LDA #$0C
0337      STA ICCOM,X
0338      JMP CIOV
0339 OPEN.1
0340      STA ICBAH,X
0341      TYA
0342      STA ICBAL,X
0343      LDA #3
0344      STA ICCOM,X
0345 LBL:040 JSR CIOV
0346      BMI EXIT
0347      RTS
0348 EXIT JSR CLOSE.CH1
0349      JMP (DOSVEC)
0350 ;
0351 ;Picture fader routine
0352 ;
0353 MSKTBL
0354      .BYTE $01,$08,$40,$02
0355      .BYTE $10,$80,$04,$20
0356 FADEIN
0357      LDA SAVMSC
0358      STA N41
0359      STA N54
0360      LDA #0
0361      STA CTR2
0362 LBL:104 LDA #7
0363      CMP CTR2
0364      BCS LBL:087
0365      RTS
0366 LBL:087 LDX CTR2
0367      LDA #0
0368      STA LBL:080
0369 LBL:090 LDA #$FE
0370      CMP LBL:080
0371      BCC LBL:088
0372      TXA
0373      PHA
0374      LDX LBL:080
0375      LDY LBL:078,X
0376      PLA
0377      TAX
0378      JSR LBL:089
0379      INC LBL:080
0380      JMP LBL:090
0381 LBL:088 TXA
0382      PHA
0383      LDX #$FF
0384      LDY LBL:078,X
0385      PLA
0386      TAX
0387      JSR LBL:089
0388      LDA LBL:091
0389      BMI LBL:092
0390      LDY CTR2
0391      CPY #2
0392      BNE LBL:093
0393      LDA SVCOLR-1,Y
0394      STA COLOR0
0395 LBL:093
0396      CPY #3
0397      BNE LBL:096
0398      LDA SVCOLR-1,Y
0399      STA COLOR1
0400 LBL:096
0401      CPY #4
0402      BNE LBL:099
0403      LDA SVCOLR-1,Y
0404      STA COLOR2
0405 LBL:099
0406      CPY #5
0407      BNE LBL:092
0408      LDA SVCOLR
0409      STA COLOR4
0410 LBL:092 INC CTR2
0411      JMP LBL:104
0412 LBL:089
0413      LDA LBL:105
0414      STA N42
0415      LDA LBL:106
0416      STA N55
0417      LDA #0
0418      STA LBL:107
0419 LBL:113 LDA #$1D
0420      CMP LBL:107
0421      BCS LBL:108
0422      RTS
0423 LBL:108 LDA (N41),Y
0424      AND MSKTBL,X
0425      BNE LBL:111
0426      LDA MSKTBL,X
0427      EOR #$FF
0428      AND (N54),Y
0429      JMP LBL:112
0430 LBL:111 ORA (N54),Y
0431 LBL:112 STA (N54),Y
0432      INC N42
0433      INC N55
0434      INX
0435      TXA
0436      AND #7
0437      TAX
0438      INC LBL:107
0439      JMP LBL:113
0440 ;
0441 ;Program Entry
0442 ;
0443 START
0444      LDA DRNUM   ;User alterable
0445      STA DDVC+1  ;drive #
0446      LDX #1      ;Tells system
0447      STX BOOT    ;no Coldstart
0448      STX PICTYPE ;and initializes
0449      DEX
0450      STX COLDST
0451      STX SOUNDR  ;No disk sound
0452 ;
0453 ;Build random table for fading
0454 ;picture.
0455 ;
```

continued on next page

```
0456 LBL:079 TXA
0457      STA LBL:078,X
0458      INX
0459      BNE LBL:079
0460      STX LBL:080
0461 LBL:081
0462      LDA #$FF
0463      STA N60
0464      EOR LBL:080
0465      PHA
0466      STA N59
0467 LBL:062 ASL A
0468      BCS LBL:061
0469      LSR N60
0470      JMP LBL:062
0471 LBL:061 LDA RANDOM
0472      AND N60
0473      CMP N59
0474      BEQ LBL:063
0475      BCS LBL:061
0476 LBL:063
0477      STA LBL:083
0478      TAX
0479      LDA LBL:078,X
0480      TAY
0481      PLA
0482      PHA
0483      TAX
0484      LDA LBL:078,X
0485      LDX LBL:083
0486      STA LBL:078,X
0487      PLA
0488      TAX
0489      TYA
0490      STA LBL:078,X
0491      INC LBL:080
0492      INC PICTYPE
0493      BNE LBL:081
0494 ;
0495 ;Build screen dislpay list by
0496 ;closing channel 6
0497 ;
0498      LDX #$60
0499      JSR CLOSE.CH6
0500 ;
0501 ;Then open GRAPHICS 8+16
0502 ;
0503      LDA #$0C
0504      STA ICAX1,X
0505      LDA #8
0506      STA ICAX2,X
0507      LDY # <SDVC
0508      LDA # >SDVC
0509      JSR OPEN.1
0510 ;
0511 ;Give screen some color and
0512 ;modify display list to
0513 ;ANTIC E (GRAPHICS 7+) display
0514 ;list
0515 ;
0516      LDA #$80
0517      STA COLOR4
0518      LDA SDLSTL
0519      STA N41
0520      LDA SDLSTH
0521      STA N42
0522      LDY #3
0523      LDA #$4E
0524      STA (N41),Y
0525      LDY #6
0526 LBL:071 LDA #$0E
0527      STA (N41),Y
0528      INY
0529      CPY #$63
0530      BNE LBL:071
0531      LDA #$4E
0532      STA (N41),Y
0533      INY
0534      INY
0535      INY
0536 LBL:073 LDA (N41),Y
0537      CMP #$0F
0538      BNE LBL:072.A
0539      LDA #$0E
0540      STA (N41),Y
0541      INY
0542      BNE LBL:073
0543 ;
0544 ;Set pointers to a hidden
0545 ;screen used to fade new
0546 ;picture into old.
0547 ;
0548 LBL:072.A
0549      STA LBL:106
0550      LDA SAVMSC+1
0551      STA LBL:106
0552      LDA LBL:106
0553      SEC
0554      SBC #$20
0555      STA LBL:105
0556      LDA LBL:105
0557      STA SAVMSC+1
0558 ;
0559 ;Set ctr to point to first
0560 ;picture.
0561 ;
0562 FIRST.PIC LDA #1
0563      STA CTR1
0564 ;
0565 ;Open the disk directory for
0566 ;reading.
0567 ;
0568 NXT.PIC
0569      JSR CLOSE.CH1
0570      LDA #6
0571      STA ICAX1,X
0572      LDA #0
0573      STA ICAX2,X
0574      LDY # <DDVC
0575      LDA # >DDVC
0576      JSR OPEN.1
0577      STY CTR2     ;Y=0 always...
0578 ;
0579 ;Read filenames 'til CTR2=CTR1.
0580 ;CTR1 is index to which
0581 ;picture was last shown.
0582 ;
0583 LBL:119 LDA CTR1
0584      CMP CTR2
0585      BCC LBL:118
0586      LDX #$10
0587      LDA # <LBL:064+1
0588      STA ICBAL,X
0589      LDA # >LBL:064+1
0590      STA ICBAH,X
0591      LDA #20
0592      STA ICBLL,X
0593      LDA #0
0594      STA ICBLH,X
0595      LDA #5
0596      STA ICCOM,X
0597      JSR CIOV
0598      BMI FIRST.PIC
0599      INC CTR2
0600      BNE LBL:119
0601 ;
0602 ;Have read the correct # of
0603 ;filenames.  Now check if
0604 ;this is a FREE SECTORS message.
0605 ;If yes, start from 1st picture.
0606 ;
0607 LBL:118
0608      JSR CLOSE.CH1
0609      LDA LBL:121
0610      CMP #$20
```

```
0611        BNE FIRST.PIC
0612 ;
0613 ;Stick "D1:" in front of
0614 ;filename.
0615 ;
0616        LDX #2
0617 NAME.LOOP LDA DDVC,X
0618        STA LBL:064,X
0619        DEX
0620        BPL NAME.LOOP
0621 ;
0622 ;Then reform name with no
0623 ;spaces, a period, the extender,
0624 ;and an end-of-line.
0625 ;
0626 NAME.LOOP.1 INX
0627        CPX #8
0628        BEQ NAME.LOOP.2
0629        LDA LBL:125,X
0630        CMP #$20
0631        BNE NAME.LOOP.1
0632 NAME.LOOP.2 LDY LBL:127
0633        LDA #'.
0634        STA LBL:125,X
0635        TYA
0636        INX
0637        STA LBL:125,X
0638        STA PICTYPE
0639        LDA #'I
0640        INX
0641        STA LBL:125,X
0642        LDA #'C
0643        INX
0644        STA LBL:125,X
0645        LDA #EOL
0646        INX
0647        STA LBL:125,X
0648 ;
0649 ;Open the picture for reading.
0650 ;
0651        JSR CLOSE.CH1
0652        LDA #4
0653        STA ICAX1,X
0654        LDA #0
0655        STA ICAX2,X
0656        LDY # <LBL:064
0657        LDA # >LBL:064
0658        JSR OPEN.1
0659        LDA LBL:105
0660        STA SAVMSC+1
0661        LDA #7
0662        LDX #$10
0663        STA ICCOM,X
0664 ;
0665 ;If the extender is 'PIC' then
0666 ;go to compressed picture load
0667 ;routine.
0668 ;
0669        LDY PICTYPE
0670        CPY #'P
0671        BNE MIC
0672        JMP PIC.LDR
0673 ;
0674 ;Otherwise, do standard load.
0675 ;
0676 MIC
0677        LDA SAVMSC
0678        STA ICBAL,X
0679        LDA SAVMSC+1
0680        STA ICBAH,X
0681        LDA #0
0682        STA ICBLL,X
0683        LDA #$1E
0684        STA ICBLH,X
0685        JSR LBL:040
0686        JSR GET.BYTE
0687        STA SVCOLR
0688        JSR GET.BYTE
0689        STA SVCOLR+1
0690        JSR GET.BYTE
0691        STA SVCOLR+2
0692        JSR GET.BYTE
0693        STA SVCOLR+3
0694 ;
0695 ;Both load types continue here.
0696 ;Fade new picture in.  Pause
0697 ;and read console keys.
0698 ;
0699 MIC.1
0700        JSR FADEIN
0701        JSR CLOSE.CH1
0702        LDA #0
0703        STA ATRACT
0704        STA RTCLOK+1
0705 LBL:134 LDA RTCLOK+1
0706        CMP WAIT
0707        BCS LBL:130
0708 LBL:132 LDA CONSOL
0709        CMP #5
0710        BNE LBL:133
0711        JMP EXIT
0712 LBL:133 CMP #3
0713        BNE LBL:133.1
0714 LBL:133.2
0715        LDA #6
0716 LBL:133.3
0717        CMP CONSOL
0718        BNE LBL:133.3
0719 LBL:133.1
0720        CMP #6
0721        BNE LBL:134
0722 LBL:130 INC CTR1
0723        JMP NXT.PIC
0724 DDVC
0725        .BYTE "D1:*.?IC",EOL
0726 SDVC
0727        .BYTE "S:",EOL
0728        *=  RUNADR
0729        .WORD START
0730        .SET 1,0
```

game of the month

can you survive 15 levels of laser death?

ARENA RACER

Article on page 49.

LISTING 1

```
GG 1 REM ARENA RACER
XP 2 REM BY J. SUTHERLAND
OK 3 REM ANTIC MAGAZINE
II 5 GOTO 1000
NU 10 SOUND 3,TM,10,12:TM=TM-0.6+0.02*(15-L):IF TM<30 THEN X=1:Y=60:SOUND 3,0,0,0:GOTO 190
JG 15 A=USR(1536,ADR(S$)+70*Y+X):IF  NOT STRIG(0) THEN FOR T=1 TO 3:NEXT T:GOTO 10
LH 20 SOUND 3,0,0,0:S=STICK(0):IF S=15 OR S=OS THEN 40
YC 30 OS=S:NYD=(S=13)+(S=9)+(S=5)-(S=14)-(S=10)-(S=6):NXD=(S=7)+(S=6)+(S=5)-(S=10)-(S=9)-(S=11)
QH 40 IF PEEK(207)<>0 THEN P=PEEK(207):POKE 207,0:GOTO 80
SL 50 IF PEEK(SL+NXD+20*NYD)<>65 THEN XD=NXD:YD=NYD
CF 60 X=X+XD:Y=Y+YD
RO 70 GOTO 10
KV 80 OS=15:IF P=65 THEN NXD=-XD:NYD=-YD:X=X-XD:Y=Y-YD:GOTO 10
IE 90 IF P=133 THEN NXD=-XD:NYD=-YD:X=X-XD:Y=Y-YD:GOTO 10
PD 100 IF P=134 THEN NXD=-XD:NYD=-YD:X=X-XD:Y=Y-YD:GOTO 10
CJ 110 IF P=131 OR P=132 THEN 190
GN 120 IF P<>200 THEN 10
CH 130 FOR J=100 TO 130:SOUND 1,J,10,12:NEXT J:SOUND 1,0,0,0
SH 140 S$(70*(Y+4)+X+9)="♥":G=G+1:SC=SC+25+2*L:POSITION 2*G-1,16:? #6;"◤":POSITION 8,14:? #6;SC:IF G<4 THEN 10
XD 150 FOR J=240 TO 125 STEP -0.5:SOUND 1,J,10,14:SOUND 2,J+1,10,14:NEXT J:SOUND 1,0,0,0:SOUND 2,0,0,0
UZ 160 SC=SC+2*INT(TM):L=L+1:MEN=MEN+(BN>3)-(MEN>10):BN=BN+1:IF BN=5 THEN BN=0
WU 170 IF L>14 THEN L=0:IF PEEK(1684)>4 THEN POKE 1684,PEEK(1684)-2
GL 180 GOSUB 1080:GOTO 10
IA 190 FOR J=9 TO 20:FOR C=9 TO 13:POKE 1671,C:SOUND 0,200,8,20-J:A=USR(1536,ADR(S$)+70*Y+X):NEXT C
BQ 200 NEXT J:POKE 1671,2:POSITION MEN*2-3*(MEN>1),18:? #6;" "
YV 210 MEN=MEN-1:IF MEN<1 THEN 250
XA 220 SOUND 0,0,0,0:FOR T=1 TO 400:NEXT T:TM=254
EK 230 S$(70*(Y+4)+X+9)="♥":GOTO 10
QZ 240 GOTO 50
XW 250 POSITION 5,17:? #6;"GAME OVER":POSITION 3,19:? #6;"press trigger":POKE 40410,0:FOR J=0 TO 3
FY 260 SOUND J,0,0,0:NEXT J
UT 270 IF STRIG(0) THEN 270
UQ 280 CLR :RUN
AW 1000 CLR :GRAPHICS 17:POSITION 4,3:? #6;"ARENA  racer":POSITION 9,5:? #6;"by":POSITION 4,7:? #6;"J.SUTHERLAND"
OE 1010 CB=PEEK(106)-6:POKE 756,CB:ML=1536:POSITION 5,9:? #6;"one moment":SETCOLOR 2,9,4
ZQ 1020 DIM T$(70),T2$(70),S$(4900)
XP 1030 T$="h▣♥▤♥▣▣♥▂▤♥▣▣♥ ▣▣♥▣▣◆":T$(9)=CHR$(CB):T$(15)=CHR$(CB+1):A=USR(ADR(T$)):RESTORE 5000
UK 1040 READ A:IF A>-1 THEN POKE ML+K,A:K=K+1:CM=CM+A:GOTO 1040
IF 1050 ML=PEEK(89)*256+PEEK(88)+3:SL=ML+87:HB=INT(SL/256):LB=SL-HB*256:POKE 1551,HB:POKE 1550,ML-HB*256
NO 1055 POKE 1667,HB:POKE 1666,LB:POKE 1674,HB:POKE 1673,LB
DJ 1060 FOR J=0 TO 111:READ A:POKE CB*256+J,A:NEXT J
DH 1070 MEN=4:L=0:BN=0
FI 1080 RESTORE 6000+L*20:READ T2$:POSITION 2,9:? #6;"ENTERING LEVEL ";L+1:G=0:TM=254
LL 1090 S$="":FOR J=1 TO 70:A=ASC(T2$(J,J))-65:RESTORE 2000+A*10
JU 1100 READ T$:S$(LEN(S$)+1)=T$:SOUND 0,120,10,12:POSITION 1,11:? #6;70-J;"  ":SOUND 0,0,0,0:NEXT J
TP 1110 X=1:Y=60:XD=0:YD=0:NXD=0:NYD=0:? #6;"↖"
IP 1120 RESTORE 6000+L*20+10:FOR J=1 TO 4:READ A:S$(A,A)=CHR$(200):NEXT J:POSITION 1,14:? #6;"SCORE: ";SC
ER 1130 POSITION 1,18:IF MEN>1 THEN FOR J=1 TO MEN-1:? #6;"▣ ";:NEXT J
UK 1140 POSITION 4,22:? #6;"LEVEL: ";L+1:GOTO 10
VB 2000 DATA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
IE 2010 DATA AAAAA♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥AAAAA
DH 2020 DATA AAAAA♥AAAA♥AAAAA♥AAAAA♥AAAAA♥AAAAA♥AAAAA♥AAAAA♥AAAAA♥AAAAA♥AAAAA♥AAAAA
PK 2030 DATA AAAAA♥♥A♥♥♥♥♥A♥♥♥♥♥A♥♥♥♥♥A♥♥♥♥♥A♥♥♥♥♥A♥♥♥♥♥A♥♥♥♥♥A♥♥♥♥♥A♥♥♥♥♥A♥♥♥AAAAA
JK 2040 DATA AAAAA♥♥♥AA▜♥♥♥♥♥♥♥♥♥♥♥♥♥♥A♥♥♥♥♥♥♥◢AA▜♥♥♥♥♥♥♥♥♥♥A♥A♥♥♥♥♥♥♥♥♥◢AA♥♥♥AAAAA
CD 2050 DATA AAAAAAAAAAA♥AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA♥AAAAAAAAAAA
RY 2060 DATA AAAAA♥♥AA▜♥♥A♥♥◢AA▜♥♥A♥♥◢AA▜♥♥A♥♥◢AA▜♥♥A♥♥◢AA▜♥♥A♥♥◢AA▜♥♥A♥♥◢AA♥♥AAAAA
ER 5000 DATA 104,104,133,204,104,133,203
DV 5010 DATA 162,0,160,1
PM 5020 DATA 177,203,157,131,157
WR 5030 DATA 201,131,208,16,169,0,145,203,136,177,203,201,65,240,4
BS 5040 DATA 169,131,145,203,200
AL 5050 DATA 201,132,208,20,169,0,145,203,232,200,192,15,240
SU 5060 DATA 51,177,203,201,65,240,4,169,132,145,203
QD 5080 DATA 201,133,208,12,165,206,201,0,208,6,200,169,132,145,203,136
IP 5090 DATA 201,134,208,12,165,206,201,0
```

```
   ,208,6,136,169,131,145,203,200
ZX 5100 DATA 232,200,192,15,240
FQ 5110 DATA 3,76,11,6,138,24
YW 5120 DATA 105,6,170,224,180,240,19
ZC 5130 DATA 165,203,24,233,185,144,2
FU 5140 DATA 230,204,165,203,24,105,70
IU 5150 DATA 133,203,76,9,6,173,218,157,1
   33,207,169,2,141
NR 5160 DATA 218,157,169,0,133,77,230,206
   ,165,206,201,12,208,4,169,0,133,206,23
   8,199,2,96,-1
NQ 5170 DATA 0,0,0,0,0,0,0,0,255,255
GN 5180 DATA 255,255,255,255,255,255
FC 5190 DATA 60,24,189,231,231,189,24,60
GB 5200 DATA 0,0,0,85,42,0,0,0
GE 5210 DATA 0,0,0,85,42,0,0,0
TF 5220 DATA 192,96,248,159,184,240,96,19
   2
NI 5230 DATA 3,6,31,249,29,15,6,3
PZ 5240 DATA 32,1,18,8,20,74,128,2
QS 5250 DATA 28,42,93,127,93,42,28,0
OO 5260 DATA 16,32,133,18,160,8,68,1
EW 5270 DATA 22,1,20,136,17,2,40,129
SV 5280 DATA 129,32,0,2,80,0,9,0
KH 5290 DATA 1,64,8,0,0,32,2,0
PP 5300 DATA 32,1,18,8,20,74,128,2
GL 6000 DATA AAAAABBBBBBEBBBBBBBEBBBBBBBE
   BBBBBEBBBBBBBEBBBBBBBEBBBBBBBEBBBBBBBA
   AAAA
QX 6010 DATA 672,1338,2075,3043
FJ 6020 DATA AAAAABEBEBBEBBBEEBBCBBBEBBEB
   BBEBECEBBEBBBBCBBBEBBCBBBEBEBBBBEBBEBA
   AAAA
GP 6030 DATA 652,1290,2222,3113
TO 6040 DATA AAAAABBCBEBGBCBFBECDCCEGCCEF
   BCECDEBEFBCBECBBEDECBBBDBEBBCEBCDEBFBA
   AAAA
QL 6050 DATA 721,1025,3034,3169
WU 6060 DATA AAAAABBCBEBBBCBFBECBCCBGCCBF
   BCBBBEBEFBCBEBBBEDBCBBBDBEBBCBBCDBBFBA
   AAAA
TN 6070 DATA 924,2042,3536,2129
RW 6080 DATA AAAAABBBBEEEBBBEEEBBBEEEBBBE
   EEBBBEEEBBBEEEBBFBBEEEBBBEEEBBBEEEBBBA
   AAAA
JE 6090 DATA 764,1290,2222,3113
NI 6100 DATA AAAAABFFFFGBGFFFEBGFFFFBFFFF
   FBFFFFFBFFBFEBEFFFFBFFBGBFBCEDDEFFFFBA
   AAAA
QF 6110 DATA 811,1324,4108,2914
SD 6120 DATA AAAAABFGGGGFFFGGDGGCCCECEGEE
   GBBBBBEGBBBBBFEFGBGGEEBBBBBEFEFGFGFBBA
   AAAA
FQ 6130 DATA 987,1115,3479,3194
GH 6140 DATA AAAAABGEGCBGBBCBDBEGBBDBBCBG
   EBBBBGBBDBCBBGBBDDBBGEGBBCBBBGBDBBBGBA
   AAAA
BZ 6150 DATA 512,1213,3314,501
MW 6160 DATA AAAAABBBGBFFEBBFFGFFBBBBEGEB
   BBGEGBBGGGGDEEEEBBFBBFFBEBBCCECBBBFBBA
   AAAA
GY 6170 DATA 512,1131,3104,851
CO 6180 DATA AAAAABBDDDDDBEBBGBGEGBDDDBBF
   BEGBGEBBBEEEBDDEDDBBCBBCBBDBBCBBEEGEBA
   AAAA
JB 6190 DATA 513,1214,3104,851
FS 6200 DATA AAAAABBEEBFBGGBDBEEBBBGGBEBB
   EEGEEBBBBBBGGEGGBBBFFDDGGBBBFEFBBGEBBA
   AAAA
SB 6210 DATA 717,1115,3479,3104
JK 6220 DATA AAAAABBBBBEGEGEBFDFCDFBEGCGE
   BBBBFFFBEEGEEBBDDDBBCCCBBEGBBGEBBEGBBA
   AAAA
FM 6230 DATA 987,1115,3759,3166
GC 6240 DATA AAAAABBBFFFFBEEEEBDDDBGGGBCC
   CBEGGEBEGEGBBBFFFBCDCCECFFBBDBCCBEGEBA
   AAAA
LW 6250 DATA 582,1283,3314,521
QC 6260 DATA AAAAABBBEEEEBEEEGEEEEBEEEBEE
   GBBBBDDDCCGEEDEEEEEFFBCEGEDBBCCEBBBBBA
   AAAA
AQ 6270 DATA 539,1502,2345,3199
KI 6280 DATA AAAAABBBBBEBBBBBEGBFBGEBFFBC
   CBDDBGEBDBEGBBFBEEBGGBEGGBBBCCBBDDBBBA
   AAAA
KQ 6290 DATA 498,2639,2905,597
```

bonus game

rapid maze game in ACTION!

AMAZING

Article on page 55.

LISTING 1

```
;     AMAZING
;     BY DAVID PLOTKIN
;     ANTIC MAGAZINE

MODULE

CARD SCRLOC=88,HIMEM=$2E5,
     PM_BASEADR,ADRES,ADRESB,SCORE=[0]

INT  DIRX=[2],DIRY=[0],XDIR,YDIR

INT ARRAY PXDR=[0 0 0 0],
          PYDR=[0 0 0 0]

BYTE T=$DA,VCOUNT=$D40B,
     PMHITCLR=$D01E,DMACTL=$22F,
     GRACTL=$D01D,PMBASE=$D407,
     PRIORITY=$26F,X0,Y0,COUNT=[0],
     LV=[5],FT=[150],CD=[20],
     PCLRM=711,COLR0=708,LOUD=[0],
     COLR1=709,COLR2=710,COLR4=712,
     FATE=53770,CURSH=752,
     TXTROW=656,TXTCOL=657,LVL=[1],
     SND1=$D20F,SND2=$D208

BYTE ARRAY YLOCL(80),
     YLOCH(80),RSH2(160),
     PMHPOS(8)=$D000,
     PX(4)=[0 0 0 0],PY(4)=[0 0 0 0],
     BEGX(4)=[0 52 52 196],
```

continued on next page

```
     BEGY(4)=[0 38 166 38],
     PM_WIDTH(5)=$D008,PLPTR,
     PM_MISMASK(4)=[$FC $F3 $CF $3F],
     PCOLR(4)=704,PMTOPF(8)=$D000,
     PMTOP(8)=$D008,PFCOL(8),PCOL(8)
BYTE ARRAY BM(0)=[$C0 $30 $C $3],
           CM(0)=[$0 $55 $AA $FF],
  CHMP1(0)=[0 0 129 129 66 66 36 36
  24 24 24 24 36 36 66 66 129 129 0 0],
  CHMP2(0)=[0 0 129 129 66 66 60 36
  36 36 36 36 36 60 66 66 129 129 0 0],
  CRT(0)=[0 0 129 129 129 195 90 126
  126 165 165 126 126 90 195 129 129
  129 0 0],
  MSTATUS(0)=[0 0 0 0],ESTAT(4),
  MX(0)=[0 0 0 0],MY(0)=[0 0 0 0],
  BLK(0)=['U'U'U'U'Z'▨'Y'e'Z'▨'Y'e'U
        'U'U'U];WIDTH=2,HEIGHT=8
BYTE ARRAY LINE,DUM
BYTE LOW=LINE,HIGH=LINE+1

PROC DLAY(CARD WAIT)
CARD COUNT
FOR COUNT=0 TO WAIT DO OD RETURN

PROC INIT7()
BYTE LOW1,HIGH1,I   CARD SCREEN=LOW1
GRAPHICS(7) COLR0=44 COLR1=196
COLR2=106 COLR4=0 SCREEN=SCRLOC I=0
WHILE I<80 DO YLOCL(I)=LOW1
YLOCH(I)=HIGH1 SCREEN=SCREEN+40 I=I+1
OD
I=0 WHILE I<160 DO RSH2(I)=I RSH 2
I=I+1
OD
RETURN

INT FUNC HSTICK(BYTE PORT)
BYTE ARRAY PORTS(4)=$278
INT ARRAY VALUE(4)=[0 1 $FFFF 0]
RETURN (VALUE((PORTS(PORT)&$C) RSH 2))

INT FUNC VSTICK(BYTE PORT)
BYTE ARRAY PORTS(4)=$278
INT ARRAY VALUE(4)=[0 1 $FFFF 0]
RETURN (VALUE(PORTS(PORT)&3))

PROC UPDATE()
TXTROW=1 TXTCOL=12 PRINTC(SCORE)
RETURN

PROC UPDATESHIP()
BYTE LOOP5
TXTROW=1
FOR LOOP5=1 TO 5 DO TXTCOL=31+LOOP5
IF LV>=LOOP5 THEN PRINT("•")
   ELSE PRINT(" ")
FI OD RETURN

PROC DRAW7(BYTE X,Y,CLR)
BYTE X1=$A0,Y1=$A1,CLR1=$A2
LOW=YLOCL(Y1)
HIGH=YLOCH(Y1)
T=RSH2(X1)
LINE(T)=(((BM(X1&3)!$FF)&LINE(T))%
        (BM(X1&3)&CM(CLR1)))
RETURN

PROC FASTDRAW(BYTE ARRAY PICTURE
        BYTE WIDTH,HEIGHT,XX,YY)
BYTE LCTR1,LCTR2   CARD LCTR3
FOR LCTR1=0 TO HEIGHT-1
DO LOW=YLOCL(YY+LCTR1) HIGH=YLOCH(YY+LCTR1)
 LCTR2=XX+WIDTH
 LCTR3=(LCTR1+1)*WIDTH-1
 DO
 LINE(LCTR2)=PICTURE(LCTR3)
 LCTR3==-1 LCTR2==-1
 UNTIL LCTR2=XX
 OD
OD RETURN

PROC PMGRAPHICS()
ZERO(PMHPOS,8)
ZERO(PM_WIDTH,5)
DMACTL=$3E PCOLR(0)=52
PM_BASEADR=(HIMEM-$800)&$F800
PMBASE=PM_BASEADR RSH 8
HIMEM=PM_BASEADR+768
PRIORITY==&$C0%17 GRACTL=3
RETURN

CARD FUNC PMADR(BYTE N)
IF N>=4 THEN N=0 ELSE N==+1 FI
RETURN(PM_BASEADR+768+(N*$100))

PROC PMCLEAR(BYTE N)
CARD CTR
BYTE ARRAY PLAYADR
PLAYADR=PMADR(N)
IF N<4 THEN ZERO(PLAYADR,$100)
 ELSE N==-4
FOR CTR=0 TO $100-1
DO PLAYADR(CTR)==&PM_MISMASK(N) OD
FI
RETURN

PROC WINDOW()
BYTE LOOP5
TXTROW=0 TXTCOL=0 CURSH=1
PRINT
("┌──────────────────────────────────────┐")
FOR LOOP5=1 TO 2 DO
TXTROW=LOOP5 TXTCOL=0 PRINT("|")
TXTCOL=38 PRINT("|")
OD  TXTROW=3 TXTCOL=0
PRINT
("└──────────────────────────────────────┘")
TXTROW=1 TXTCOL=3 PRINT("SCORE: ")
UPDATE() TXTCOL=20 PRINT("MEN LEFT: ")
UPDATESHIP()
RETURN

PROC MOVEIT(BYTE ARRAY SHAPE BYTE
           WHICH,NUM,XX,YY)
ADRES=PMADR(WHICH)+YY
MOVEBLOCK(ADRES,SHAPE,NUM)
PMHPOS(WHICH)=XX
RETURN

PROC PUTMAN()
BYTE LP
FOR LP=0 TO 3 DO
MSTATUS(LP)=0 ESTAT(LP)=0 OD
X0=120 Y0=102 MOVEIT(CHMP1,0,20,X0,Y0)
FOR LP=1 TO 3 DO
PX(LP)=BEGX(LP) PY(LP)=BEGY(LP)
MOVEIT(CRT,LP,20,PX(LP),PY(LP)) OD
RETURN

PROC BORDER()
BYTE L1,L2
FOR L1=0 TO 159 DO
  FOR L2=0 TO 3 DO
  DRAW7(L1,L2,1) DRAW7(L1,L2+76,1)
OD OD
FOR L1=0 TO 79 DO
  FOR L2=0 TO 3 DO
  DRAW7(L2,L1,1) DRAW7(L2+156,L1,1)
OD OD
RETURN

PROC DOTS()
```

```
BYTE L1,L2
FOR L2=8 TO 72 STEP 16 DO
  FOR L1=8 TO 156 STEP 8 DO
DRAW7(L1,L2,3) OD OD
FOR L2=16 TO 72 STEP 16 DO
  FOR L1=8 TO 156 STEP 16 DO
DRAW7(L1,L2,3) OD OD
RETURN

PROC BOARDDRAW()
BYTE L1,L2
BORDER()
FOR L1=2 TO 36 STEP 4 DO
  FOR L2=12 TO 68 STEP 16 DO
  FASTDRAW(BLK,2,8,L1,L2)OD OD
DOTS()
RETURN

PROC TESTCOL()
BYTE LL
FOR LL=0 TO 7 DO
PFCOL(LL)=0 PCOL(LL)=0 OD
DO UNTIL VCOUNT&128 OD
FOR LL=0 TO 7 DO
PFCOL(LL)=PMTOPF(LL)
PCOL(LL)=PMTOP(LL) OD
PMHITCLR=1
RETURN

BYTE FUNC PMHIT(BYTE N,CNUM)
IF N<4 THEN N==+4 ELSE N==-4 FI
IF CNUM<4 THEN
 RETURN((PCOL(N) RSH CNUM)&1)
 ELSE CNUM==&3
 RETURN((PFCOL(N) RSH CNUM)&1)
FI RETURN(0)

BYTE FUNC LLOC(BYTE XX,YY,CLR)
BYTE X1=$A0,Y1=$A1,CLR1=$A2,L1,L2
LOW=YLOCL(Y1) HIGH=YLOCH(Y1)
T=RSH2(X1) L1=X1&3
L2=LINE(T)&BM(L1)
IF (L2&CM(CLR1))=(BM(L1)&CM(CLR1))THEN
RETURN(1) FI;SOMETHING THERE
RETURN(0)

BYTE FUNC LKAHD(INT XD,YD BYTE XX,YY)
BYTE XA,YA,XB,YB,RS1,RS2
XA=XX-48 YA=(YY-32) RSH 1
IF XD>0 THEN XA==+7+XD XB=XA
 YA==+1 YB=YA+7
 ELSEIF XD<0 THEN XA==+XD XB=XA
 YA==+1 YB=YA+7
 ELSEIF YD>0 THEN XB=XA+7
 YA==+9 YB=YA
 ELSEIF YD<0 THEN XB=XA+7 YB=YA
 ELSE RETURN(0)
FI RS1=LLOC(XA,YA,1) RS2=LLOC(XB,YB,1)
IF RS1+RS2=0 THEN RETURN(1)
   ELSE RETURN(0);OK
FI RETURN(0);BLOCKED

PROC NEWLEVEL()
BYTE LL
SNDRST() SCORE==+COUNT*LVL
UPDATE() COUNT=0 LVL==+1
FOR LL=0 TO 7 DO PMCLEAR(LL) OD
DOTS() PUTMAN()
DIRX=0 DIRY=0
IF LVL<11 THEN FT==-10 CD==+10 FI
RETURN

PROC MSLDROP(INT XD,YD)
BYTE TRIG=644,XA,YA,LP,MASK,LD=[0],TT=[0]
IF LD>1 THEN LD==-2 FI
SOUND(1,LD LSH 3,10,LD)
IF TRIG=1 THEN TT=0 FI
IF TRIG=1 OR (XD=0 AND YD=0) OR TT=1
 THEN RETURN FI
FOR LP=0 TO 3 DO
IF MSTATUS(LP)=0 THEN MSTATUS(LP)=1
 IF XD>0 THEN XA=X0 YA=Y0+9
   ELSEIF XD<0 THEN XA=X0+7 YA=Y0+9
   ELSEIF YD>0 THEN XA=X0+4 YA=Y0
   ELSE XA=X0+4 YA=Y0+18
 FI MASK=PM_MISMASK(LP)!$FF LD=12 TT=1
 MY(LP)=YA MX(LP)=XA
 PLPTR(MY(LP))==%MASK
 PLPTR(MY(LP)+1)==%MASK
 PMHPOS(LP+4)=MX(LP) EXIT
FI OD RETURN

PROC MSLGET()
BYTE LP,LD1=[0]
IF LD1>1 THEN LD1==-2 FI
SOUND(2,LD1 LSH 4,10,LD1)
FOR LP=0 TO 3 DO
IF PMHIT(LP+4,0)=1 THEN
MSTATUS(LP)=0 LD1=12
PLPTR(MY(LP))==&PM_MISMASK(LP)
PLPTR(MY(LP)+1)==&PM_MISMASK(LP)
PMHPOS(LP+4)=0 EXIT FI OD RETURN

PROC GOTBUMPED()
BYTE LQ,LD2=[0],LQ1
IF LD2>0 THEN LD2==-1 FI
SOUND(3,LD2 LSH 3,8,LD2)
FOR LQ=0 TO 3 DO FOR LQ1=1 TO 3 DO
IF PMHIT(LQ+4,LQ1)=1 THEN
 LD2=14 ESTAT(LQ1)=1 MSTATUS(LQ)=0
 PLPTR(MY(LQ))==&PM_MISMASK(LQ)
 PLPTR(MY(LQ)+1)==&PM_MISMASK(LQ)
 PMHPOS(LQ+4)=0
FI OD OD
FOR LQ=1 TO 3 DO
IF ESTAT(LQ)>0 THEN ESTAT(LQ)==+1
 PCOLR(LQ)=FATE
FI
IF ESTAT(LQ)=FT THEN ESTAT(LQ)=0
 PMCLEAR(LQ)
 PCOLR(LQ)=(RAND(15) LSH 4)+6
 PX(LQ)=BEGX(LQ) PY(LQ)=BEGY(LQ)
 MOVEIT(CRT,LQ,20,PX(LQ),PY(LQ))
FI OD RETURN

PROC MUNCH()
BYTE TIME=20,X1,Y1
IF LOUD>1 THEN LOUD==-2 FI
SOUND(0,8,LOUD LSH 3,LOUD)
IF PMHIT(0,10)=0 THEN DLAY(1) RETURN FI
LOUD=12 X1=X0-48 Y1=(Y0-32) RSH 1
DRAW7(X1+3,Y1+4,0) DRAW7(X1+3,Y1+5,0)
DRAW7(X1+4,Y1+4,0) DRAW7(X1+4,Y1+5,0)
COUNT==+1
IF COUNT=135 THEN NEWLEVEL() FI
RETURN

PROC CHANGEDIR(BYTE WH)
BYTE F,LP
IF FATE<CD THEN F=RAND(4)
 IF F=0 THEN PXDR(WH)=2 PYDR(WH)=0
  ELSEIF F=1 THEN PXDR(WH)=-2
  PYDR(WH)=0 ELSEIF F=2 THEN
  PXDR(WH)=0 PYDR(WH)=2 ELSE
  PXDR(WH)=0 PYDR(WH)=-2
 FI
FI
IF LKAHD(PXDR(WH),PYDR(WH),PX(WH),
 PY(WH))=0 THEN PXDR(WH)==-PXDR(WH)
```

continued on next page

```
 PYDR(WH)==-PYDR(WH)
FI RETURN

PROC SMARTS(BYTE WHICH)
BYTE X,Y
X=PX(WHICH) Y=PY(WHICH)
IF (X=52 OR X=68 OR X=84 OR X=100
 OR X=116 OR X=132 OR X=148 OR X=164
 OR X=180 OR X=196) AND (Y=38 OR Y=70
 OR Y=102 OR Y=134 OR Y=166) THEN
 CHANGEDIR(WHICH)
FI RETURN

PROC ENDGAME()
BYTE TRIG=644,ST=755,TIME=20
SCORE==+COUNT*LVL PMHITCLR=0 UPDATE()
COUNT=0 LVL=1 TXTROW=2 TXTCOL=8
PRINT("GAME OVER  PRESS FIRE")
DO ST=(TIME RSH 4)&1 UNTIL TRIG=0 OD
LV=5 UPDATESHIP()
SCORE=0 TXTROW=1 TXTCOL=12
PRINT("      ") TXTROW=2 TXTCOL=8
PRINT("                    ")
UPDATE() DOTS() PUTMAN() FT=150 CD=20
XDIR=0 YDIR=0 DIRX=0 DIRY=0 ST=0
RETURN

PROC OUCH()
BYTE LC,LD
IF PCOL(4)=0 THEN RETURN FI
LC=Y0+10  LD=Y0+10
DO LD==+2 IF LD>200 THEN LD=200 FI
LC==-2 IF LC<30 THEN LC=30 FI
IF (LC=30 AND LD=200) THEN EXIT FI
SOUND(0,LC,8,8) SOUND(1,LD,8,8)
DUM(LC)=FATE DUM(LD)=FATE
DLAY(250) DLAY(250) DLAY(250)
OD SNDRST()
FOR LC=0 TO 7 DO PMCLEAR(LC) OD
LV==-1 UPDATESHIP()
IF LV=0 THEN ENDGAME() ELSE PUTMAN()
PMHITCLR=0  FI RETURN

PROC CHASE()
BYTE LP
FOR LP=1 TO 3 DO SMARTS(LP)
PX(LP)==+PXDR(LP) PY(LP)==+PYDR(LP)
IF ESTAT(LP)=0 THEN
MOVEIT(CRT,LP,20,PX(LP),PY(LP)) FI
OD RETURN

PROC MOVEMAN()
BYTE STCK=632,TIME=20
XDIR=HSTICK(0) LSH 1
YDIR=VSTICK(0) LSH 1
IF XDIR<>0 AND YDIR<>0 THEN YDIR=0 FI
IF STCK=15 THEN XDIR=DIRX YDIR=DIRY FI
IF LKAHD(XDIR,YDIR,X0,Y0)=1 THEN
 X0==+XDIR
 Y0==+YDIR DIRX=XDIR DIRY=YDIR ELSEIF
 LKAHD(DIRX,DIRY,X0,Y0)=1 THEN
 X0==+DIRX Y0==+DIRY
 ELSE DIRX=0 DIRY=0
FI MOVEIT(CHMP1,0,20,X0,Y0)
RETURN

PROC MAIN()
BYTE XX,COUNT,TIMER=20,ATRACT=$4D
SND1=3 SND2=0
INIT7() PMGRAPHICS() PCLRM=50
PLPTR=PMADR(4) DUM=PMADR(0)
FOR XX=0 TO 7 DO PMCLEAR(XX) OD
FOR XX=1 TO 3 DO
PCOLR(XX)=(RAND(15) LSH 4)+6     OD
WINDOW() BOARDDRAW() PUTMAN() ENDGAME()
DO TESTCOL() MUNCH() MOVEMAN() OUCH()
MSLGET() CHASE() MSLDROP(DIRX,DIRY)
ATRACT=0 GOTBUMPED() OD
RETURN
```

communications

automatic log-on program

TSCOPE AUTODIALER

Article on page 13.

LISTING 1

```
GO 10 REM AUTODIAL.BAS
KZ 20 REM BY CHARLES JACKSON
RH 30 REM ANTIC MAGAZINE
KL 40 GRAPHICS 0:POKE 710,100:POKE 709,12
FZ 50 DIM NUM$(15),ACNUM$(20),PW$(25)
XL 60 ? ,"          TSCOPE          "
FJ 70 ? ," AUTODIAL FILEMAKER "
BK 80 ? :? ,,"by C. Jackson"
SM 90 ? :? :? "Phone number";:INPUT NUM$
AK 100 ? :? "Access Number";:INPUT ACNUM$
WB 110 ? :? "Password";:INPUT PW$:POKE 71
   0,66
CB 120 ? :? "Insert TSCOPE disk.":? "Pres
   s [START] to write AUTODIAL.SYS"
ZM 130 POKE 53279,8
VL 140 IF PEEK(53279)<>6 THEN 140
DL 150 CLOSE #1:OPEN #1,8,0,"D:AUTODIAL.S
   YS"
SL 160 ? #1;NUM$
EJ 170 ? #1;"^C]:";ACNUM$
UH 180 ? #1;"]:";PW$
LP 190 CLOSE #1
PY 200 POKE 710,0:? "↖AUTODIAL.SYS file c
   reated.":?
MG 210 TRAP 250
CJ 220 OPEN #1,4,0,"D:TSCOPE.OBJ":CLOSE #
   1
NQ 230 ? "Remember to change the name of
   your"
GP 240 ? "TSCOPE.OBJ file to AUTORUN.SYS.
   "
OB 250 END
```

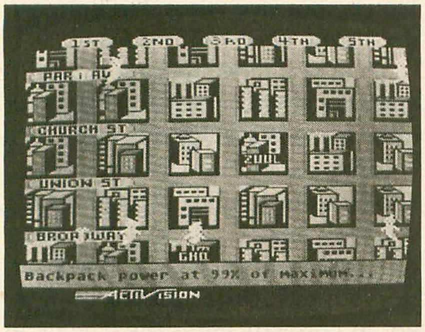
product reviews

GHOSTBUSTERS

Activision
2350 Bayshore Frontage Rd.
Mountain View, CA 94043
(415) 960-0410
$29.95—48K disk

Reviewed by Harvey Bernstein

The marriage between hit movies and computer software has been a rocky one in the past. Games based on the cinema have rarely been commercially or artistically successful. That's usually because the game is produced as a rush-job to capitalize on the motion picture's success. Until now, that is. . .



Ghostbusters from Activision, is the first adaptation to capture both the feel and the theme of the movie on which it is based. For those one or two **Antic** readers who haven't seen the movie, I'll explain.

Supernatural phenomena (referred to in the game as PK levels) in YOUR town are on the rise and ghosts are everywhere. As the owner of the local Ghostbuster franchise, it is up to you to sweep the streets for mobile ghosts (Roamers), clean all haunted buildings of their inhabitants (Slimers), and finally face down the dreaded Marshmallow Monstrosity at the Temple of Zuul. Succeed, and fame and fortune are yours. Fail, and bankruptcy awaits.

Of course, no ghostbusters worth their salt can go into business without the proper equipment, and you have the option to buy Image Intensifiers, PK Energy Detectors, Ghost Traps, Bait, etc.

As a new franchisee, the bank supplies you with $10,000 to start. But as you progress and earn more money, you can buy more sophisticated equipment. You can win at Ghostbusters by finishing the game with more money than you started. But sneaking two men into the Temple of Zuul will earn you a substantial bonus.

This is Activision's first attempt at a role-playing game, and while the game is enjoyable, there is a flaw in the design. At the end of a game, if you are successful, you are given an account number to correspond to your name and winnings. It is up to you to make a record of this number, and enter it again next time you want to play. Any deviation in the number or in spelling your character's name, and you must start over from the beginning. It should have been a simple matter to put in a save-game routine to simplify matters.

And then there's the music. While the adaptation of Ray Parker Jr.'s hit is well done, it plays throughout the game, over and over again. Since a typical game may last 15–20 minutes, a way to toggle the music off would be more than appreciated. As it is, I've taken to playing Ghostbusters with the monitor sound turned all the way down.

But these are just minor complaints. Ghostbusters is most enjoyable to play, and I hope it's a sign of what Atari owners can expect from Activision in the future.

SPY VS. SPY

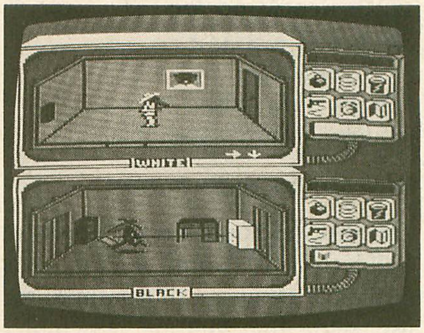
First Star Software
22 E. 41 Street
New York, NY 10017
(212) 532-4666
$29.95, 48K—disk

Reviewed by Harvey Bernstein

Too many products being released these days seem to be rehashes of the same tired arcade themes. So it gives me great pleasure to announce that **Spy Vs. Spy** is one of the most original and clever games for Atari computers yet.

The Black and White secret agents, created by Cuban cartoonist Antonio Prohias, have been one of the most popular features in Mad Magazine since 1960. The game, with an excellent Atari adaptation by ace programmer Jim Nangano, not only remains faithful to the cartoons, but is challenging and great fun to play.

As the White spy, you race the clock and your opponent Black (controlled by either another player or the computer) to find 5 items hidden



within an embassy. Once you acquire the briefcase, secret plans, key, passport and money, you must find your way through a maze of rooms to the exit leading to the airport.

But that's not all! During play, you and your opponent leave traps for each other—bombs, electrified water, guns with strings attached to the trigger and so forth. Setting off one of these booby-traps puts you out of commission for several valuable seconds, giving your opponent the edge.

Of course, as a well-armed spy, you have an arsenal of remedies at your disposal. So the umbrella neutralizes the electrified water, the scissors saves you from the gun with the string, etc.

One of the most unique features of Spy vs. Spy is a technique First Star calls Simulvision. This splits the screen in half, so that the activities of

continued on next page

product reviews

White can be seen in the top half, and Black in the bottom, allowing each player to see what the other is up to. When a player enters a room already occupied, the action shifts to one half of the screen for a winner-take-all brawl.

I cannot recommend this game highly enough. The graphics and animation exploit all the possibilities of the Atari. And with several levels of play, Spy vs. Spy should provide loads of fun for both novices and experienced gamesters.

MAC/65 TOOLKIT

Optimized Systems Software, Inc.
1221B Kentwood Ave.
San Jose, CA 95129
(408) 446-3099
$39.95, 16K—disk,
requires MAC/65 Assembler

Reviewed by Andy Barton

The **MAC/65 Toolkit** is an impressive collection of some 67 macros (assembly language subroutines) for use with the MAC/65 Assembler Editor. These macros greatly enchance the speed and ease of assembly language programming for both the novice and the experienced programmer.

The Toolkit's macro calls mimic many BASIC and assembly language commands. This makes an assembly language program almost as easy as a BASIC program to write and debug.

The macros are grouped into three libraries (files). The first library is a collection of utility routines for graphics, math, I/O and program control.

The second library offers 11 macros for setting up single line resolution Player/Missile graphics, moving the players and missiles with a vertical blank interrupt, and detecting collisions.

The third library offers a VBI routine for vertical, horizontal, and diagonal fine scrolling over a large screen display using the joystick.

The Toolkit allows assembly language beginners to focus on overall programming without having to develop complicated routines. An elementary understanding (or handy reference book) of assembly language is necessary for using this kit. It is also desirable to have a moderate familiarity with Atari's P/M graphics. The user's manual is clear and concise, but it's not as helpful for newcomers as I would have liked.

The P/M graphics library needs a macro for joystick input. Writing one might be a good first project for the user. The joystick routine from the scroll library, while not directly transferable, is a good starting point.

The libraries use fairly large blocks of memory. The utility library itself occupies slightly over six pages (about 1 1/2K). The P/M graphics library occupies a little less then two pages and the scroll library just over one page of memory. If memory space becomes a problem, you can, with a bit of effort, go through the specific libraries deleting any unused macro before final assembly.

U. S. ADVENTURE

First Star Software
18 E. 41st Street
New York, NY 10017
(212) 532–4666
$29.95—48K disk

Reviewed by Anita Malnig

This learning game might have some trouble competing with the latest Infocom adventure. However, **U. S. Adventure**—by **Antic** contributing editor Jerry White—could work very well in a history class, or be enjoyed by a youthful history buff. You've got to know your facts to succeed.

First you've got to know the order in which each state entered the union. There's a help key to give you clues, but each clue takes away points. You must move from state 1 to state 2, etc., by using directional signals which appear on the screen in the form of a compass.

After you've correctly guessed the state, you choose your next move from an Options Menu. From this menu you can choose Time Travel, Take Event, Review Map (here's where you get clues to the order of states), and several other less-used options.

Take Event and Time Travel test your knowledge of American history some more. You choose Take Event only after you have correctly chosen the next state's entrance. You're given several historical events and must weed out ones that may be bogus. Watch out for those! They can drastically alter your points.

Then you move to Time Travel to guess the year that the particular event took place. Time Travel offers nice computer sound and graphics as you appear to be looking through a long colorful tunnel. Years, 1776, 1821, etc., pass by and you control when to stop, advance, or go backward. You've got to correctly guess the date of an event with as little time travel as possible. (I hadn't read the instructions all the way through and got very intrigued with making those years go backward and forward through this tunnel ranging in hues from yellows to purples to blues. Well, I paid for the fun with my score!)

This learning game is full of interesting facts and proves to be a good history lesson. However, the instructions are not easy to follow and there are a lot of them. Getting from state to state seemed more convoluted than it had to be. I also found a spelling error: Massachesettes. That's really unacceptable in any piece of software, and especially in a learning game.

However, none of this is enough to turn thumbs down on the whole program. Young history buffs will enjoy U. S. Adventure and the game could certainly add a spark to any classroom history lesson.

product reviews

50 MISSION CRUSH

Strategic Simulations, Inc.
883 Stierlin Road, Building A-200
Mountain View, CA 94043
(415) 946-1200
40K—disk, requires BASIC
$39.95

Reviewed by Karl Wiegers

50 Mission Crush puts you in the pilot's seat of a B-17 heavy bomber in World War II. Your goal is to survive 50 missions from an Air Force base in England against 23 targets in Nazi-occupied Europe.

Your opposition includes enemy fighters and flak guns, weather, your own inexperience, and the random number generator. This role-playing game lets you share the feelings of a real pilot—relief when a "milk run" target is selected by the computer, dismay when yet another fighter shoots holes in your damaged bomber, frustration when the target is protected by clouds, anxiety as you pray your fuel will last until you return to England.

You control the movement of the bomber as well as functions such as dropping bombs, changing altitude, and fighting fires. You direct the fire of your machine guns when fighters appear. And you watch helplessly as puffs of flak appear around the plane. The crew members become more effective at their jobs as they gain experience. Games and crews can be stored on disk for continuation at another time.

This is not a visually exciting game. The few animation sequences used are very simple. The game moves slowly in spots, possibly because it is written in BASIC. Combat sound effects are good, but more sound features would add to the game. The game is easy to learn and play. A typical mission takes 5 to 10 minutes of real time.

The strength of 50 Mission Crush lies in its detailed simulation of combat results. Damage accrues gradually and realistically. Consumption of fuel and ammunition require constant decision-making. I took more damage from flak than from enemy fighters, in contrast to the historical reality.

Unfortunately, there is not much of a learning curve with 50 Mission Crush. Random events play a larger role in your fate than do skill and practice. This is a good operational-level war game, but don't expect a lot of exciting air combat action.

BEYOND CASTLE WOLFENSTEIN

Muse Software
347 N. Charles Street
Baltimore, MD 21201
(301) 659-7212
$34.95, 32K—disk

Reviewed by Harvey Bernstein

When Muse Software introduced **Castle Wolfenstein** for the Apple in 1981, it quickly shot to the top of the charts and remained there as one of the most popular games for any microcomputer. The Atari translation was remarkably faithful to the Apple, right down to the lousy sound and black-white-green-purple graphics.

Now we have the sequel, **Beyond Castle Wolfenstein**, and while there have been some minor improvements, the game play doesn't provide nearly as much depth as the documentation suggests.

The scenario in the follow-up is different, yet similar enough to the original to allow the same spare graphics. As an allied intelligence agent, you must penetrate Der Führer's bunker, a 3-level maze of rooms. Hidden in a closet on the first level is a time bomb, which you must find and then set outside Hitler's office, two levels below, after which you must retrace your steps out before the whole place goes up.

As in the original, each room is swarming with unfriendly guards. You have no uniform to allow you access, but you do have numbered passes. When you enter a room, the guard demands to see your pass. If you show the wrong one, you will probably be arrested, but you do have money to bribe the guards.

Once you find the correct pass for a level, it works with every guard on that level, so the game becomes a lot easier. The chests of the previous game have been replaced by closets, some of which are locked, requiring the talents of a safecracker.

There are some improvements over the original, most notably the speech synthesis used for the guards. With a little practice, you can recognize their grunts as actual German words. Also, if you accidently walk into a wall, you don't get the filling-rattling routine that accomanied Castle Wolfenstein.

> The game promises more than it delivers in strategy.

Now for the bad news. The game promises more than it delivers in strategy. For example, while I found keys in several closets, after playing 3 games in progressively difficult levels, I found nothing to use them on. Also, there is a toolkit which the documentation says can be used to disable the alarm system. Not only do I still not know how to disable the alarm, but I've yet to figure out why I would want to.

Once you know which passes to use, you can breeze through the game with only mapping needed. It's this sameness and ease that keeps me from going back to play Beyond Castle Wolfenstein again and again. Not to mention that it takes so long to load that it recalls fond memories of my old 410 recorder.

SHOPPER'S GUIDE

service center

Service Centers, Retailers, to get your listing in Antic call (415)661-3400

ALABAMA

VIDEO REPAIR CO.
2009 CENTERPOINT RD.
BIRMINGHAM
205-854-5212

RAINBOW CITY SERVICE CENTER
244 RAINBOW PLAZA
GADSEN
205-442-6810

C & R ELECTRONICS
704 HOLCOMBE AVE.
MOBILE
205-473-3030

BUSINESS SERVICES
2828 CHESTNUT ST.
MONTGOMERY
205-834-2290

ARIZONA

RICK'S TV & APPLIANCE
1104 E. DEUCE OF CLUBS
SHOW LOW
602-537-7625

CALIFORNIA

LEARNING TREE COMPUTER CTR.
2441 N. TUSTIN SUITE BCD
SANTA ANA
714-667-1575

COMPUTER SUPPORT SERVICE
52 S. LINDEN AVE. #1
SOUTH SAN FRANCISCO
415-589-9800

D & G COMPUTERS
4156 MANZANITA AVE. #200
CARMICHAEL
916-485-7779

SAN JOSE COMPUTER
1844-E ALMADEN RD.
SAN JOSE
408-723-2025

B & C COMPUTERVISION
3400 EL CAMINO REAL
SANTA CLARA
408-554-0666

ATCOM COMPUTERS
1421 THOUSAND OAKS BLVD.
THOUSAND OAKS
805-497-1220

AUTHORIZED COMPUTER SERVICE
951 W. FOOTHILL BLVD.
UPLAND
714-985-2101

TESTEK
7224 VALJEAN AVE.
VAN NUYS
818-786-6890

COMPUTER JUNCTION, INC.
15000 7TH ST. SUITE 214
VICTORVILLE
619-245-3622

COLORADO

AMERICAN TELEVISION SERVICE CO
1226 W. LITTLETON BLVD.
LITTLETON
303-795-2040

LOOKING GLASS MICROPRODUCTS
4233 WEST EISENHOWER
LOVELAND
303-669-2681

FLORIDA

R & S ELECTRONICS, INC.
3245 W. MCNAB RD.
FT. LAUDERDALE
305-979-6763

BOURQUE'S ELECTRONIC SERVICE
180 RICHPIEN RD.
FT. WALTON BEACH
904-862-3346

MR. SOFTWARE
101 HOLLYWOOD FASHION CENTER
HOLLYWOOD
305-981-9090

ENTERTAINMENT SERVICES ELECT.
811 EDGEWOOD AVE. S.
JACKSONVILLE
904-786-1305

COMPUTER IMAGE
10061 SUNSET DR.
MIAMI
305-271-1224

COMP-U-PHONE
6160 EDGEWATER DR.
SUITE E.
ORLANDO
305-291-1712

GEORGIA

GOODWIN'S ELECTRONICS
JUNCTION SHOPPING CENTER
ALBANY
912-435-9605

NORMAN'S ELECTRONICS, INC.
4014 PEACHTREE RD. NE
ATLANTA
404-237-3349

ELECTRONIC SPECIALISTS
1685 EDNA PLACE
MACON
912-742-5628

RADIOS AND MICROCOMPUTERS INC.
3833A WASHINGTON RD
MARTINEZ
404-863-9071

HARRIS TV
1 E. MONTGOMERY CROSSROAD
SAVANNAH
912-927-2084

HAWAII

KONA COMPUTER
75-5706 HANAMA PL. #107
KAILUA-KONA
808-329-8574

ILLINOIS

DIGITAL WORLD, INC.
711 ARMY TRAIL RD.
ADDISON
312-543-9000

OMEGA ENTERPRISES
7823 N. 2ND ST.
ROCKFORD
815-282-1477

INDIANA

WRIGHT ELECTRONICS
614 N. MAIN ST.
EVANSVILLE
812-423-2845

CITIZEN'S TV & VIDEO
827 W. GLENPARK AVE.
GRIFFITH
219-924-0047

THE COMPUTER CORNER
7101 BROADWAY
MERRILLVILLE
219-738-3282

KANSAS

MIDWEST APPLIANCE SERVICE
METCALF SOUTH MALL
OVERLAND PARK
913-341-6688

KENTUCKY

FACTORY ELECTRONICS
2422 PALUMBO DR.
LEXINGTON
606-269-7341

VANOVER TV HOSPITAL
2027 CUMBERLAND AVE.
MIDDLESBORO
606-248-3827

LOUISIANA

COMPUTER ELECTRONICS
1955 DALLAS DR.
BATON ROUGE
504-924-8066

MASSACHUSETTS

CUSTOM ELECTRONICS
238 EXCHANGE ST.
CHICOPEE
413-592-4761

BEACON TV-ELECTRONICS
8 TILESTON ST.
EVERETT
617-389-8600

CONDOR SERVICE
372 SOUTH MAIN ST.
SHARON
617-784-2382

ROCOM, INC.
184 MAIN ST.
WAREHAM
617-295-2542

MARYLAND

NATIONAL BUSINESS & SECURITY
8639 LOCH RAVEN BLVD.
BALTIMORE
301-665-8870

EDGEWOOD TV & AUDIO
4932 EDGEWOOD RD.
COLLEGE PARK
301-441-9116

CROFTON TV & VIDEO SERVICE
2217 DEFENSE HWY
CROFTON
301-721-1700

COMPUVISION COMPUTER CENTER
6445 BURWOOD PL.
GLEN BURNIE
301-850-4055

ADVANCED COMPUTER SERVICE CO.
287 E. GREEN ST.
WESTMINSTER
301-876-8202

MAINE

AUTOMATIC SERVICES
22 TARGET INDUSTRIAL CIRCLE
BANGOR
207-942-6769

J.D. ELECTRONICS
385 STEVENS AVE.
PORTLAND
207-775-1411

MICHIGAN

FUTURE DIRECTIONS
1520 N. VAN DYKE
BAD AXE
517-269-7211

THE FAMILY COMPUTER CENTER
3895 W. 12 MILE RD.
BERKLEY
313-543-0520

CHASE TRANSISTOR SERVICE
521 LEONARD ST. NW
GRAND RAPIDS
616-454-9000

ABL ELECTRONIC SERVICE, INC.
32 E. 14 MILE ROAD
MADISON HEIGHTS
313-588-6663

SOLID STATE SERVICE
548 SHATTUCK RD.
SAGINAW
517-752-0606

MINNESOTA

PHILIPS COMMUNICATIONS & TV
748 N.E. HARDING ST.
MINNEAPOLIS
612-378-7200

USER FRIENDLY COMPUTER
8465 PLAZA BLVD.
SPRING LAKE PARK
612-786-8181

MISSOURI

SOUTHLAND ELECTRONICS
22 E. OLIVE DR.
AURORA
417-678-4623

MIDWEST APPLIANCE SERVICE
METRO NORTH MALL
KANSAS CITY
816-436-7010

COMMUNITY SOUND & VIDEO
1834 S. STEWART
SPRINGFIELD
417-887-3391

A & E ELECTRONICS CORP
2001 BIG BEND BLVD.
ST. LOUIS
314-645-7733

J & S VIDEO VISIONS
1051 WASHINGTON SQUARE CENTER
WASHINGTON
314-239-2677

MISSISSIPPI

ELECTRONIC SERVICES
2315 25TH AVE
GULFPORT
601-863-3772

NORTH CAROLINA

SOUTHERN PHOTO TECHNICAL SVCE
2610 SOUTH BLVD.
CHARLOTTE
704-523-0012

NEW JERSEY

DEPENDABLE PARTS, INC.
168 MAIN ST.
CHATHAM
201-635-5888

S R DATA
2141 WOODBRIDGE AVE.
EDISON
201-985-5017

ADVANCED ELECTRONICS
VILLAGE SHOPPING CENTER
RT.206
FLANDERS
201-584-1252

TELSAR ELECTRONIC SERVICES, INC
829 STONE RD.
LAUREL SPRINGS
609-783-8500

VIDEO ELECTRONICS COMPUTER SVC
1418 ROSELLE ST.
LINDEN
201-925-1418

VIDEO CONNECTION OF SOMERSET
900 EASTON AVE.
SOMERSET
201-545-8733

K & S ELECTRONICS
119 HAMILTON BLVD.
SOUTH PLAINFIELD
201-755-4204

NEW YORK

ISLAND VIDEO COMPUTER SERVICE
35 MIDDLE COUNTRY RD.
CORAM
516-736-1001

AARDVARK ELECTRONIC SERVICE
44 CASTLE ST.
GENEVA
315-789-5295

LONG ISLAND COMPUTER GENERAL
103 ATLANTIC AVE.
LYNBROOK
516-887-1500

ABC ELECTRONICS SERVICE CO.
392 THURSTON RD.
ROCHESTER
716-328-1840

OHIO

ARJAY MICRO
1385 BETHEL RD.
COLUMBUS
614-459-4219

STATION FUN TV, INC.
206 CLINTON ST.
DEFIANCE
419-782-8545

COMPUTER CREATIONS
424 E. STROOP RD.
KETTERING
513-294-0222

B AND G ELECTRONICS, INC.
15729 MADISON AVE.
LAKEWOOD
216-521-2855

ERIC MARTIN'S, INC.
5485 WARRENSVILLE CENTER RD.
MAPLE HGTS
216-663-2032

VIDEO COMPUTER WORLD, INC.
2223 WOODVILLE RD.
OREGON
419-691-7282

DODD CO.
7795 W. RIDGEWOOD DR.
PARMA
216-886-2828

OREGON

NORTHWEST COMPUTER SUPPORT,INC
10200 S.W. NIMBUS G-1
PORTLAND
503-684-3280

PENNSYLVANIA

KIBLER'S INDEPENDENT TV
526 FALLOWFIELD AVE.
CHARLEROI
412-483-7484

PARK T. MORROW, INC.
627 W. 26TH ST.
ERIE
814-455-7566

DEBUG BYTES COMPUTERS
662 PHILADELPHIA ST.
INDIANA
412-349-7290

GRUSS ELECTRONIC REPAIR
HILLS PLAZA
JOHNSTOWN
814-266-1395

TESCO, INC.
9237 ROOSEVELT BLVD.
PHILADELPHIA
215-677-5000

service center

NATIONAL TELEVISION SERVICE
5461-63 PENN AVE
PITTSBURGH
412-361-5400

BOYD TV
719 LANCASTER AVE
WAYNE
215-688-3727

RHODE ISLAND

VIDEO ENCOUNTERS
MAINES SHOPPING CENTER
WAKEFIELD
401-783-3460

SOUTH CAROLINA

ELECTRONIC SERVICE CO.
1736 DECKER BLVD
COLUMBIA
803-782-2705

COASTAL TV & APPLIANCE CO.
603 HWY 501
CONWAY
803-248-2686

SOUTH DAKOTA

TAYLOR AUDIO-VISUAL, INC.
1009 DAKOTA S.
HURON
605-352-3205

HOUSE OF TELEVISION
601 SOUTH DULUTH AVE.
SIOUX FALLS
605-338-9051

TENNESEE

HI-FI SERVICE CENTER
4608 HIXSON PIKE
CHATTANOOGA
615-877-6781

BILL'S T.V. SALES & SERVICE
3843-C DICKERSON RD.
NASHVILLE
615-865-5000

TEXAS

TV CENTER
202 S. WILLIS
ABILENE
915-677-1171

PACIFIC STEREO
525 113TH ST.
ARLINGTON
817-640-3094

LONGS/ENTRONIX
5800 MAPLE AVE.
DALLAS
214-358-3222

COMPUTER HOME, INC.
3548 KNICKERBOCKER
SAN ANGELO
915-944-9795

UTAH

INTERWEST ELECTRONICS CORP.
4091 SOUTH STATE ST.
SALT LAKE CITY
801-266-5301

VIRGINIA

SALEM COMPUTER CENTER
4034 PLANK RD.
FREDERICK
703-786-8126

VIDEO UNLIMITED SERVICE CENTER
1707 ROUTE 17
GRAFTON
804-898-5318

L & Y ELECTRONICS
13670 JEFFERSON DAVIS HWY.
WOODBRIDGE
703-494-3444

WASHINGTON

ON LINE COMPUTERS PLUS
13710 NE 20TH ST.
BELLEVUE
206-644-2080

BUTLER'S TV & COMPUTER SERVICE
28717 PACIFIC HWY SOUTH
FEDERAL WAY
206-941-9096

ARTICULATE SYSTEMS
E 9405 SPRAGUE AVE.
SERVICE CTR
SPOKANE
509-922-0255

JOHNSON'S TELEVISION SERVICE
N 4424 WALL
SPOKANE
509-327-9566

WISCONSIN

AUTHORIZED TV
810 NINTH ST.
GREEN BAY
414-499-4215

DAN'S CITY WIDE T.V.
1259 E. JOHNSON ST.
MADISON
608-255-4144

MISTER TV SERVICE
5455 W. BURLEIGH ST.
MILWAUKEE
414-873-2415

WEST VIRGINIA

COMPUTERS PLUS, INC.
2077 CHARLESTON TOWN CENTER
CHARLESTON
304-342-4848

Antic welcomes program submissions from readers. Just send us your program and accompanying article, we'll pay you if we publish them.

We prefer to see your listing and text on both *paper and disk.*

Sending us your program on cassette is also okay. But please put program copies on *both* sides of the cassette.

Always include a stamped, self-addressed envelope so your materials can be returned.

ADVERTISERS

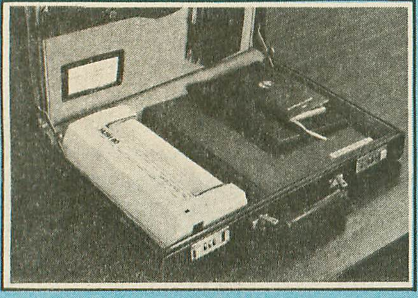
This is provided as a convenience and as a courtesy to advertisers. **ANTIC** does not guarantee accuracy or comprehensiveness.

new products

New Products notices are compiled by the **Antic** staff from information provided by the products' manufacturers. Antic welcomes such submissions, but assumes no responsibility for the accuracy of these notices or the performance of the products listed.

HUSH 80

(printer)
Ergo Systems, Inc.
1360 Willow Road
Menlo Park, CA 94025
(415) 322-ERGO
$133.99



Ergo Systems tells us this is the lowest cost 80 column dot-matrix thermal printer on the market. **HUSH 80** prints at 80 characters per second, weighs 25 ounces and fits in your briefcase, with room left over for lunch.

BOUNTY BOB STRIKES BACK

(game software)
Big Five Software
P.O. Box 9078–185
Van Nuys, CA 91409
(818) 782–6861
$49.95
16K, cartridge

After more than a three year wait, Big Five Software has finally come up with **Bounty Bob Strikes Back**, the sequel to best selling Miner 2049ER. The game (on ROM cartridge) offers more of the same comic ladder action, a spectacular high-score screen, and a price tag of $49.95. Jes' like the old days.

MASTERING YOUR ATARI THROUGH 8 BASIC PROJECTS

(book)
Prentice Hall
Englewood Cliffs, NJ 07632
(201) 592–2640
$19.95 book and disk

Written by the editors and programmers of Micro Magazine, this package claims to teach all levels of programmers BASIC techniques, while providing programs with utility and enjoyment value. Projects include a spreadsheet, music player, games, and utilities.

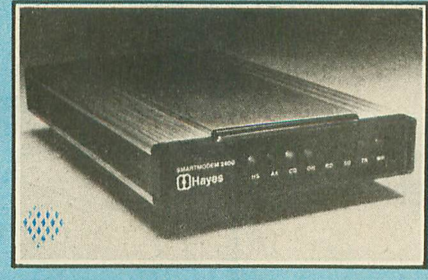
U-PRINT A64

(interface/buffer)
Digital Devices Corp.
430 Tenth Street, Suite N205
Atlanta, GA 30318
(800) 554–4898
In Georgia (404) 872–4430
$179.95—64K, $119.95—16K
$89.95—interface only

U-Print A replaces Digital's popular Ape-Face printer interface. The two higher-priced models also include memory buffers. A copy button for up to 255 multiple copies and reset button are built in.

SMARTMODEM 2400

(modem)
Hayes Microcomputer Products, Inc.
5923 Peachtree Industrial Blvd.
Norcross, GA 30092
(404) 449–8791
$899 (estimated retail)

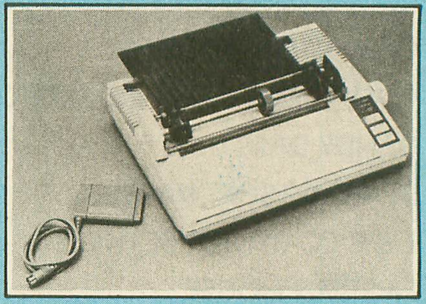


For some, 1200 baud is just not fast enough. The **Smartmodem 2400** is the new Rolls Royce of modems. There may not yet be anyone out there fast enough to receive your signal, but this will look pretty nifty next to your Atari.

HOMEWRITER 10

(printer)
Epson America
2780 Lomita Boulevard
Torrance, CA 90505
(213) 539–9140
$269 plus $60 for interface

This new Epson 80-column, dot-matrix printer is aimed at the home market. It comes with plug-in ROM interface cartridges designed for specific printers. Epson claims their **Homewriter** operates in both draft and "near letter quality". Print changes (condensed, emphasized, etc.) can be chosen from a control panel. Optional add-ons include tractor feed and cut-sheet feeder.



PRINT SHOP

(graphics software)
Broderbund Software
17 Paul Drive
San Rafael, CA 94903
(415) 479–1170
$44.95
48K disk

The long-awaited Atari version of the hit graphics design program has finally been pronounced ready by Broderbund. With **Print Shop** it's easy to make greeting cards, banners, signs, letterheads and custom stationery. Works with most of the popular dot-matrix printers.

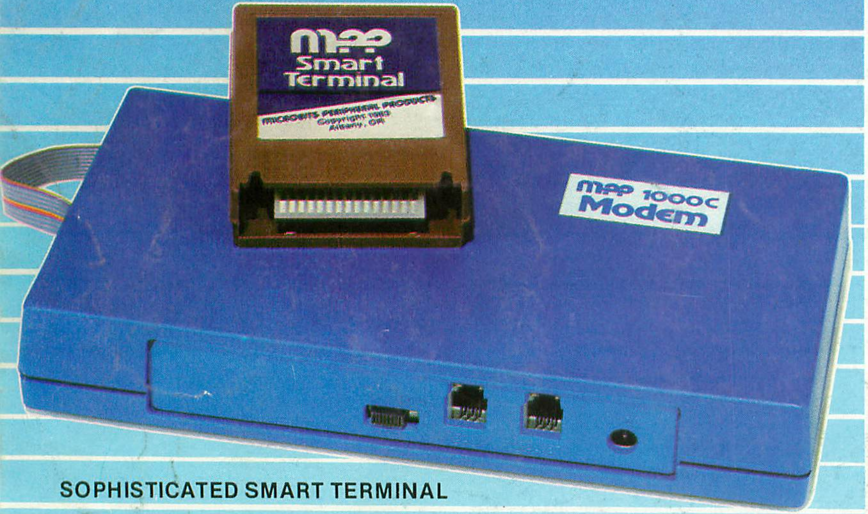
Return the favor. When you call a manufacturer or supplier about a product you've seen advertised or otherwise mentioned in ANTIC, please tell them so. This will help us to continue to bring you the latest information about products that will make your Atari computer an even more valuable investment in the future. —ANTIC ED

Expand Your Atari
...With peripherals from MPP



MicroPrint
Parallel Printer Interface

- Works with Atari,400, 800, 600XL and 800XL
- Replaces Atari 850™ Interface Module
- Compatible with all software
- 5-foot cable with Centronics plug (compatible with Epson, NEC, Prowriter, etc.)
- Connects to serial bus on computer
- 2 Year Warranty



SOPHISTICATED SMART TERMINAL SOFTWARE ON CARTRIDGE

FEATURES:

- ☐ Supports XMODEM Protocol
- ☐ ASCII/ATASCII Translation
- ☐ Allows Transfer of Files Larger than Memory
- ☐ Upload/Download of Text and Programs
- ☐ 100% Machine Language
- ☐ Loads a 65 Column Screen Driver
- ☐ Multiple Buffers
- ☐ Off-Line Editing
- ☐ Variable Baud Rate
- ☐ Parity Options
- ☐ Full/Half Duplex

ONLY
$149.95

MPP-1000C
Modem

- Auto Answer/Auto Dial
- Direct Connect to Phone Line
- No Atari 850™ Interface Module Needed
- Includes AC Adapter/Power Supply
- Free CompuServe DemoPak™
- 1 Year Warranty
- Connects to Joystick Port
- Works on **ALL** Atari Computers

MICROBITS PERIPHERAL PRODUCTS
225 Third Avenue, SW • Albany, OR 97321
ORDERS: 1-800-624-7532
CUSTOMER SERVICE: 1-503-967-9075